SEP -5

DOCUMENTATION PAGE

AFRL-SR-AR-TR-02-0344

1. REPORT DATE *(DD-MM-YYYY)* 24-08-2002	**2. REPORT TYPE** FINAL	**3. DATES COVERED** *(From - To)* 01 APR 93 - 31 MAR 96
4. TITLE AND SUBTITLE **Mechanics of Shrinkage during Processing of Carbon-Carbon Composites**		**5a. CONTRACT NUMBER** F49620-93-C-0015
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Julius Jortner		**5d. PROJECT NUMBER**
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Jortner Research & Engineering, Inc P O Box 219 Cloverdale, OR 97112		**8. PERFORMING ORGANIZATION REPORT NUMBER** JRE02002A
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dept of Air Force AFOSR 801 N Randolph St Arlington, VA 22203-1977		**10. SPONSOR/MONITOR'S ACRONYM(S)**
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Unlimited; Approved for Public Release

20021031 012

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report describes research into the mechanisms of shrinkage of resin-matrix laminates as they are carbonized during the process of making carbon-carbon composites. Activities have included: direct observation of composite samples while heating in the hot stage of a scanning electron microscope; calculation of strain fields from SEM images of fiber bundles; and development of image-analysis methods for characterizing the geometries of fiber arrays, with emphasis on quantifying contacts among neighboring fibers. Fiber-fiber contacts are found to be a major factor determining the shrinkage behavior of composites during carbonization.

15. SUBJECT TERMS

carbon/carbon composites, processing, fiber-fiber contacts, shrinkage, microstructure, carbonization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Julius Jortner
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED		61	**19b. TELEPHONE NUMBER** *(include area code)* 503-392-3001

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

FOREWORD

The work described in this report was conducted at Jortner Research & Engineering, Inc, (JR+E), in Costa Mesa, California and Cloverdale, Oregon; at Clarkson University, Potsdam, NY; and at the Southwest Research Institute (SWRI), San Antonio, Texas.

Samples of materials were provided to this program by Mr Howard Maahs and Mr Bob Yamaki of NASA Langley, and by Dr Jim Zimmer of Acurex Corp.

The principal investigator was Julius Jortner of JR+E.

Major collaborators were:

Professor Steven W Yurgartis, MIE Dept, Clarkson University, Potsdam,NY, working in the area of microstructural image analysis, and

Dr David L Davidson, SWRI, San Antonio, Texas, in whose laboratory hot-stage microscopy was performed.

Assistance was provided by

Mr Tad Guski, a graduate student of Prof Yurgartis at Clarkson University,

Mr Ben Yurgartis of Potsdam, NY, who provided some original software for visualizing the compaction of fiber cross-section arrays, and

Mr John Campbell, at SWRI, who ran the hot-stage microscope in Dr Davidson's laboratory.

Although the collaborators contributed essentially to any success this effort may have, they are not personally responsible for putting together this report. The selection of the material included here, and any errors of omission or commission, are the responsibility of the principal investigator. It is noted here with thanks that many of the essential contributions drawn upon were memos and other informal results provided by Professor Steven Yurgartis.

This report has been compiled late, several years after the end of the research effort, because of impediments that arose in the personal life of the principal investigator. Hopefully, some aspects of the work remain of interest to the technical community despite the lapse of time.

TABLE OF CONTENTS

1.0 INTRODUCTION and BACKGROUND[1]

Many C/C laminates are made from carbon-fiber prepregs, usually with a thermosetting resin. These resin prepregs are stacked, compacted and cured under pressure, and then baked to carbonize the matrix. The conversion of resin to carbon is accompanied by decreases in the mass and volume of the matrix. A slowly heated phenolic resin, for example, loses about 40 percent mass while increasing in specific gravity from about 1.25 to about 1.45; thus the volume of remaining matrix is only about 50 percent of the original matrix volume; some high-yield resins, like polyarylacetylenes (PAA), offer reduced mass loss (circa 15%), but nevertheless lose about 25-30% in volume on carbonization[2]. Shrinkage of matrix tends to change the size of the composite, usually (but not always) resulting in a decrease of composite volume. The situation is complex, involving not only restraint by the dimensionally-stable fibers, but also the evolution of pyrolysis gases and the potential release of elastic energy stored in the compacted reinforcement array.

Fiber content affects shrinkage of a carbonizing composite. For unidirectional (1D) composites, Hüttner[3] reports a more-or-less linear relation between measured shrinkage and fiber-volume fraction, over a range of fiber-volume fractions from 0.25 to 0.80. For bidirectional (2D) laminates of carbon cloth in phenolic resin, Marnoch[4] gave a linear correlation for crossply shrinkage strain, $S = a - bF$, where a and b are positive constants, and F is the fiber volume fraction in the cured state. For the same laminates, Olmsted and Harvard[5] show the linear relation fits data over $.33 < F < .69$. At the higher fiber-volume fractions, the laminates grew 2 to 3%; the shrinkage was zero at $F = .62$, approximately. Olmsted and Harvard attribute the growth to "fabric stress relief".

Shrinkage also is affected by reinforcement architecture. With tridirectional (3D) orthogonal reinforcement, composite shrinkage is effectively restrained by fibers[6]. For 2D laminates shrinkage is essentially all in the crossply direction, as the in-plane dimensions are stabilized by fibers. The crossply shrinkage of a 2D laminate should be close to the diametral shrinkage of a 1D bundle of the same fiber content; but perhaps somewhat less in woven-fabric laminates because of the constraining effect of yarn crimp in cloth.

---

[1] Illustrative charts and figures that would augment this discussion may be found in (a) J. Jortner, in Proc. 1991 Rocket Nozzle Technology Subcommittee Mtg, JANNAF, Huntsville, November 1991, CPIA Publication and (b) J. Jortner, in Extended Abstracts of 20th Bienn Conf on Carbon, Amer Carbon Soc, 1991, 402-403.

[2] H. A. Katzman, Poyarylacetylene Resin Composites, Aerospace Corp Report SSD-TR-90-013, 2 April 1990.

[3] W. Hüttner, Doctoral Dissertation, Univ. Karlsruhe, Feb 1980, p.96. See also: Z. Werkstofftech, 16, 1985, 430-435.

[4] K. Marnoch, talk at Symposium on Carbon Fibers & Composites, SUNY, Buffalo, 1988.

[5] T. W. Olmsted and R. C. Harvard, in Int'l SAMPE Tech Conf Series, vol.20, 1988 (closed sessions), 140-145. (limited distribution)

[6] J. L. Perry and D. F. Adams, Carbon, 14, 1976, 61-70.

In laminates or unidirectional tows, fiber-volume fractions higher than about 0.5 usually are obtained with the aid of compaction pressures applied before cure of the matrix resin. Most laminates intended for conversion to carbon-carbon are cured under crossply pressures greater than 1 MPa. Gutowski et al[7] measured loads necessary to compact unidirectional stacks of oil-impregnated carbon fibers, and provide a correlative model derived from the notions that "fiber network stiffness is governed by bending beam behavior of fibers between multiple contact points," and that the number of contact points increases with increasing compaction. In keeping with the elastic-beam hypothesis, their data show the deformation to be mostly reversible on release of load, with more than 10% potential increase in bundle volume on unloading from about 1.4 MPa pressure. Similar springbacks may be expected with 2D cloth laminates. Some preliminary experimentation by White and Gopalakrishnan[8] shows one plain-weave stack to spring back more than 15% in thickness on reduction of compaction pressure from about 0.7 MPa. Thus, it seems reasonable that some laminates of high fiber-volume fractions may grow on carbonization, when the carbonizing matrix softens or cracks sufficiently to allow some relief of compaction stresses frozen-in by the resin cure.

1D composites made with high-modulus non-surface-treated fibers shrink less, at the same fiber-volume fraction, than do those made with high-strength surface-treated fibers[2,9]. The same trend is seen in bi-directional cloth laminates[10]. Fitzer and Gkogkidis[11] attributed the lesser shrinkage to poorer fiber-matrix adhesion, which allows matrix to "shrink away from the fibers", leaving voids; with good adhesion, they claimed, the matrix clings to the fibers and the bundle cross-section must shrink more than if the interface voids occurred. An alternate (or supplementary) explanation, in terms of fiber-packing geometries, was proposed more recently[12] and is described below.

1.1 Fiber Contact Networks

Assuming negligible porosity in the cured condition, total composite volume (taken as unity) is $F + R$, where F is the fiber volume and R is the thermoset resin volume. After carbonization, the total volume becomes $V = F + C + P$, where F is the fiber volume (assumed unchanged), C is the carbon matrix volume, and P is the pore volume generated during carbonization. Defining m as the carbon-volume yield of the resin ($m = C/R$), we can express the volumetric strain on carbonization as $(V-1) = (m-1)(1-F) + P$. Taking m as constant, the maximum volume change occurs[10,] when $P = 0$ (Fig. 1-1).

For rigid fibers, the maximum fiber volume fraction depends on the geometry of the fiber array. For circular fibers, it is 0.907, in the limit of close-packed hexagonal array; 0.785 for a square

---

[7] T. G. Gutowski, Z. Cai, S. Bauer, D. Boucher, J. Kingery, and S. Wineman, J. Comp. Mat., 21, 1987, 650-669.
[8] J. L. White, University of California, San Diego, private correspondence, 15 October 1991.
[9] L. M. Manocha, E. Yasuda, Y. Tanabe, and S. Kimura, Carbon, 26, 1988, 333-337.
[10] Y. R. Yamaki and H. G. Maahs, in Proc. 17th Conf. Metal Matrix, Carbon and Ceramic Matrix Composites, NASA Conf Publ 3235, Part 2, 1994, 723-739.
[11] E. Fitzer and A. Gkogkidis, in Petroleum-Derived Carbons, ACS Symp 303, American Chemical Society, Washington, DC, 1986, 346ff.
[12] (a) J. Jortner, in Extended Abstracts of 20th Bienn Conf on Carbon, Amer Carbon Soc, 1991, 402-403.
(b) J. Jortner, in Proc. 1991 Rocket Nozzle Technology Subcommittee Mtg, JANNAF, Huntsville, November 1991, CPIA Publication.

array; and 0.605 for open hexagonal packing. Because fiber volume-fraction increases as the composite shrinks, the solid line in Fig. 1-1 is not valid at very high volume fractions, and upper bounds for shrinkage are given by the heavy dashed lines, for each of the three illustrated fiber arrays. The bound for hexagonal close packing was given previously by Hüttner[2]; the dashed-line bounds were described by Jortner[11]. The slopes of the limiting dashed lines depend on the value of m. At any point in the graph below the heavy limit line, porosity must be generated proportional to the vertical distance to the limit line. The relation of porosity to shrinkage is shown by the light dotted lines.

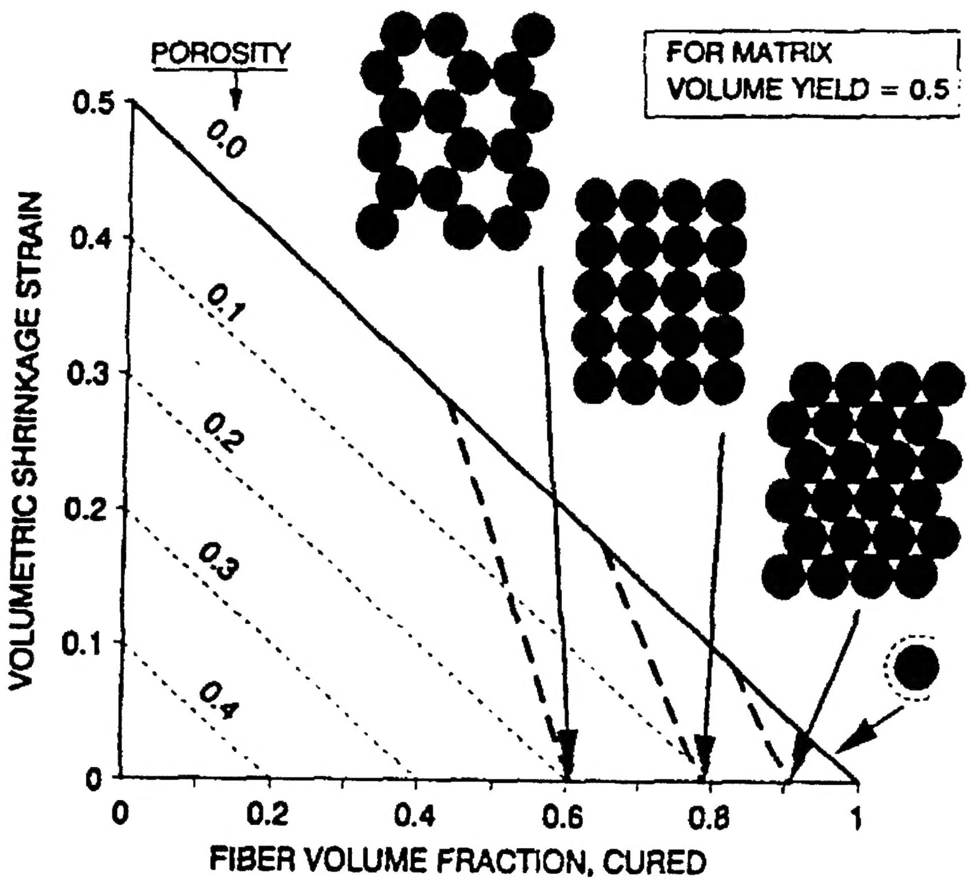


Fig. 1-1. Theoretical limits to shrinkage of unidirectional composites for a matrix carbon-volume yield of 0.5. Thick dashed lines show limits to shrinkage for several idealized fiber arrays. Dotted lines indicate porosity of the composite after carbonization.

Clearly, details of fiber packing can affect composite shrinkage. Generally, for well- aligned fibers, the composite cannot shrink (beyond an extent consistent with fiber-fiber contact deformations) when fiber packing provides a continuous network of contacting cross-sections [cf Lange et al[13] and Boriek et al[14]]. Although, any fully populated regular array would, at its limiting fiber-volume fraction, provide a contact network, such networks can exist in incompletely populated arrays[15,16]. Percolation theory[14] suggests continuous contact paths can exist at volume fractions as low as about 0.45. Two types of arrays may be imagined: *well-separated* arrays of non-contacting fibers, and *clustered* arrays (Fig.1-3). It seems clear that well-separated arrays would tend to experience greater cross- sectional shrinkage during carbonization.

To explore the potential effects of clustering, assume that fibers (on the average) agglomerate in contact clusters of an effective fiber-volume fraction F_c (within the cluster), with surrounding matrix separating such clusters. The matrix within the cluster, is isolated from the rest of the composite and cannot contribute to its overall shrinkage. At any given overall fiber-volume fraction F, the portion of matrix that contributes to composite shrinkage is $1 - F/F_c$; assuming no porosity generation in that portion, the volume strain of the composite is limited to $(V-1) = (m-1)(1 - F/F_c)$. For $F_c = 1$, representing separated individual fibers with no isolated matrix, there is no fiber contact and the composite response is that given previously as the limiting response in Fig. 1-1. As Fig. 1-2 shows, a value of unity for F_c provides a good fit to Hüttner's data for HF fibers in a PAA resin (m=.69); for $F_c = 0.65$, the estimated shrinkage turns out to be a fair fit to Hüttner's data for the less wettable M40 fibers in the same resin.

To the extent that contact decreases the fiber-matrix surface area, clustering and contact networks would seem to be promoted by a lack of wetting of fibers by matrix liquid.

1.2 Inter-Fiber Forces

The rudimentary analyses above have assumed the matrix carbon-volume yield is a constant, a characteristic property of the particular matrix precursor used. Actually, there are good reasons for distrusting this assumption. Most matrix carbons, like glassy carbons in bulk, contain closed micropores. When the matrix is severely strained, as it might be around fibers during heating, the pores might become elongated and occupy less relative volume[17]; thus, the bulk density of the carbon (and therefore the carbon-volume yield) would depend on the preferred orientation induced by anisotropic straining (or restraint) during carbonization.

Compressive deformations occur at fiber-fiber contacts during carbonization[11b]. To obtain compression along the centroidal line between neighboring fibers, it is not necessary the fibers be in contact (cf, Lange et al[18]). Two illustrative cases[11b] serve to indicate some bounding conditions for the local occurence of compression or tension between neighboring fibers. Consider two equal circular filaments of unit diameter separated by a distance s. The maximum

---

[13] F. F. Lange, L. Atteraas, and F. Zok, Acta Met, 39, 1991, 209-219.
[14] A. M. Boriek, J. E. Akin, and C. D. Armeniades, J. Comp. Mat., 22, 1988, 986-1002.
[15] R. Zallen, Physics of Amorphous Solids, Wiley, 1983. Chapters 2 and 4.
[16] S. Torquato, Appl Mech Rev, 44, Feb 1991, 37-76.
[17] G. M. Jenkins, K. Kawamura, and L. L. Ban, Proc. R. Soc. Lond. A., 327, 1972, 501-517.
[18] F. F. Lange, et al, Proc. ASM Conf. on Innovative Inorganic Comp. (1990).

shrinkage strain over the fiber center-to-center distance (without fiber shape deformation) is $e_{max} = - s/(s + 1)$. This strain would occur if, after carbonization, the fibers were touching. When the laminate's linear carbonization strain (parallel to the line joining the centers of the two fibers), e_{lam} , is such that $e_{lam} < e_{max}$ then, at some time during carbonization, there would be compressive force acting on the matrix between the two fibers, along the line joining the fiber centers. In terms of s, there would be compression between fibers when $s < -e_{lam}/(1 + e_{lam})$. On the other hand, if the matrix shrinks isotropically, with a volume yield of m, and no cracking occurs between two fibers, the strain over the fiber center-to-center distance is $e_{iso} = es/(s + 1)$, where $e = m^{(1/3)} - 1$. Then, if $e_{lam} > e_{iso}$, there would be tensile force acting on the matrix between the two fibers, along the line joining the fiber centers. This assertion relies on the intuition that matrix linear shrinkage in a certain direction, if less than the isotropic value, would require some tensile force in that direction. In terms of s, there would be tension between fibers when $s > e_{lam}e^{-1}/(1 + e_{lam}e^{-1})$. These bounds are crude, and there is an intermediate range of values of s for which they give no information. Nevertheless, they support the claim that we should expect compression across much of the inter-fiber matrix, and probably some fiber-shape deformation, wherever the fiber cross-sections are less than a certain distance apart.[19]

Compression of matrix between closely spaced filaments probably is a factor promoting its graphitizability.

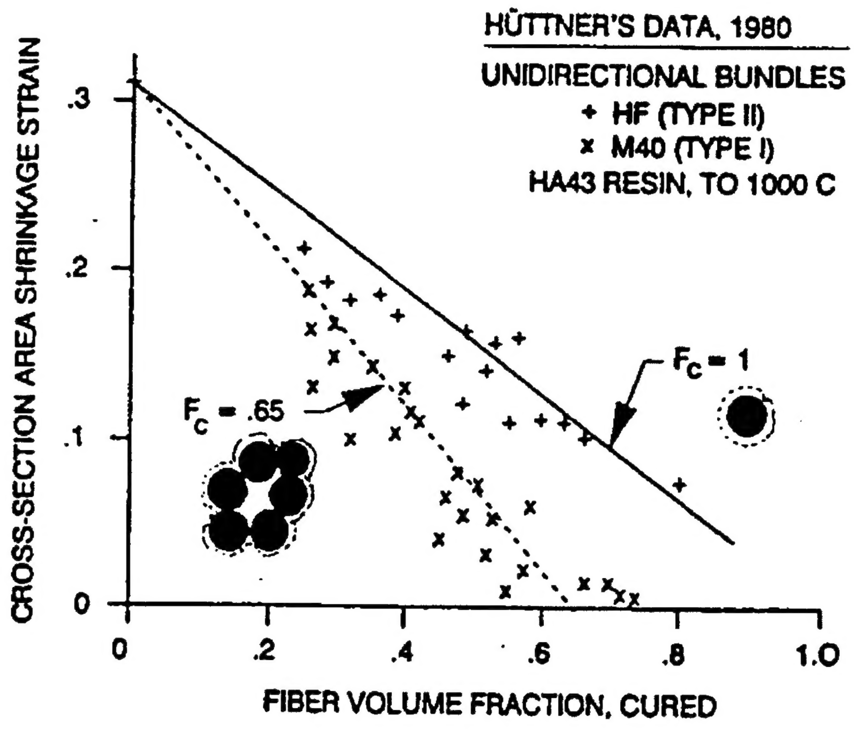


Fig. 1-2. Shrinkages measured on carbonization of unidirectional composites of two fibers in a PAA resin. Data from Hüttner, 1980. Dotted line represents trend predicted from simplified cluster-effect theory for Fc = .65. After Jortner, 1991.

[19] Examples of such deformations were shown in [11].

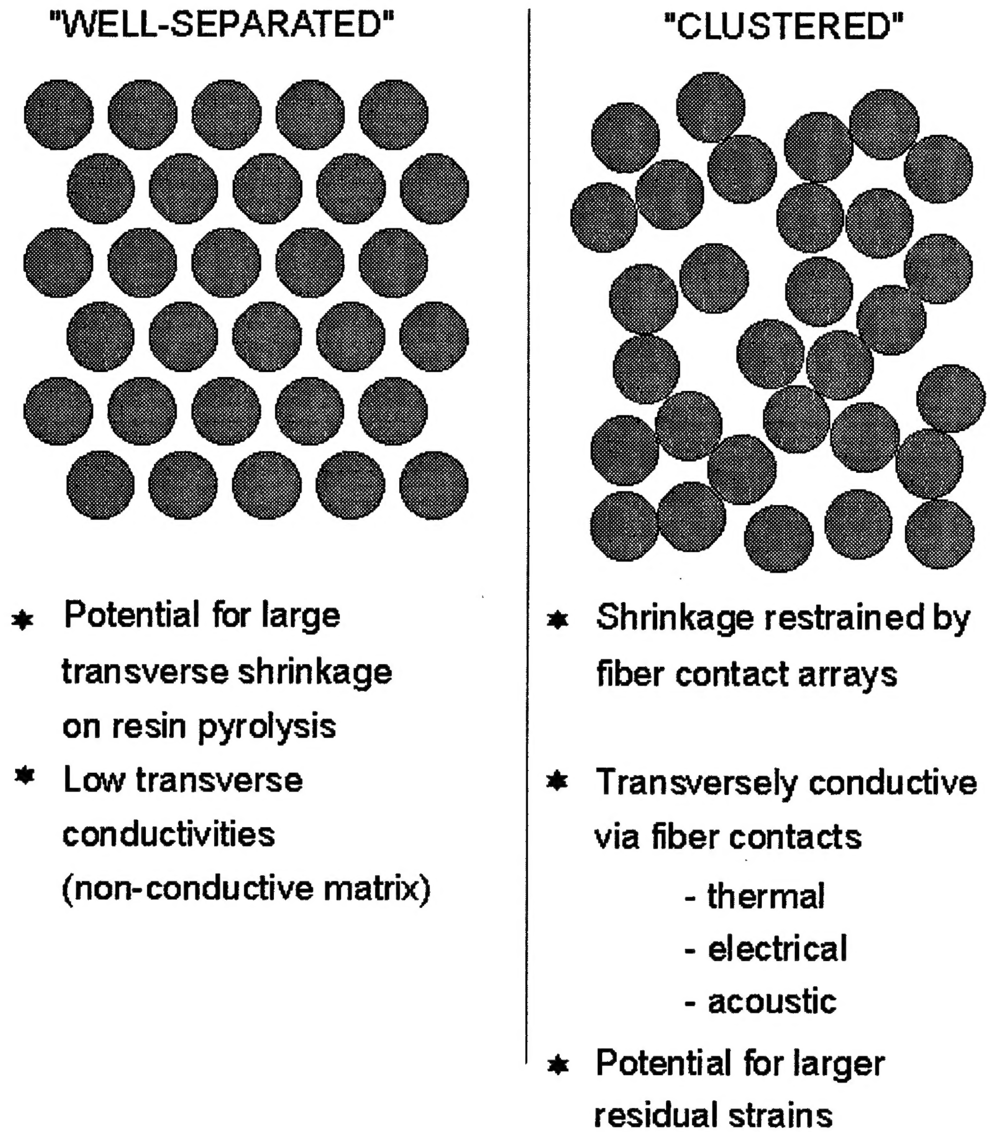


FIG. 1-3. Sketches of well-separated and clustered arrays.

1.3 Scope of Study

The research described in this report explores fiber arrays in real and model composites, combining microstructural observations and image analyses, seeking descriptions of fiber arrays that are relevant to the understanding of processing and behavior of carbon-carbon composites.

2.0 HOT STAGE MICROSCOPY

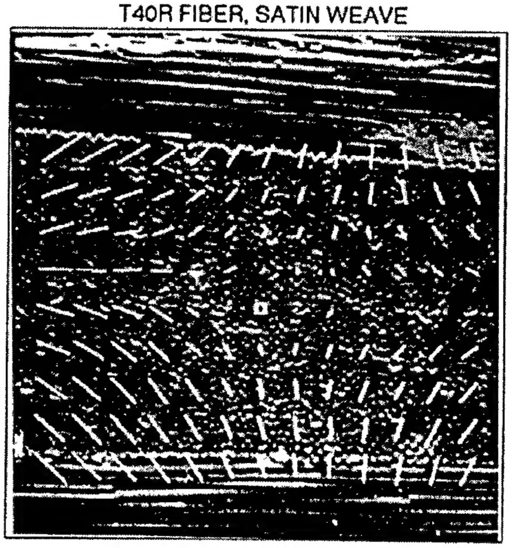
A scanning electron microscope equipped with a hot stage[20] and capable of making observations during carbonization of the viewed sample was available at the Southwest Research Institute. The system at SWRI includes a computerized image-analysis system for measuring displacements, deformations and strains[21].

Preliminary explorations of the facility's capabilities for this project were initiated early in this program. Three composites, in the phenolic-matrix state, were heated in the hot stage and photographed. Displacements were measured from photographs for two of these samples, using the computerized system, and maps of displacements were prepared.

For one sample, T40R-0.4, maps below show the displacement vectors at 590 C, as measured from the initial state at room temperature. Each set of vectors is referred to a point at the center of the photo. The displacement grids are superimposed on photos at 590 C at two magnifications. Additional detail is shown in the photos on the next page.[22]

These initial results show the SEM hot stage, with associated displacement analysis software, to

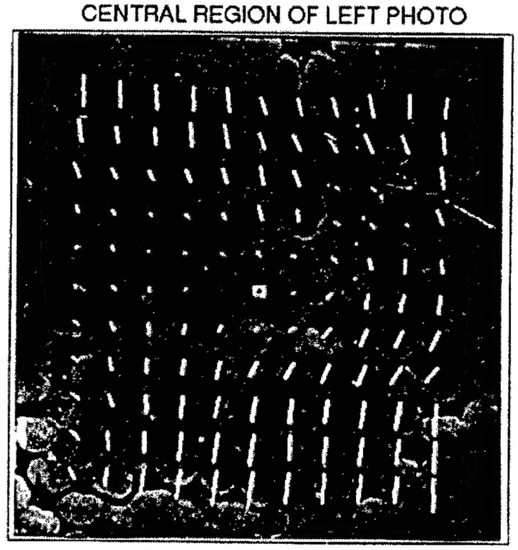
T40R FIBER, SATIN WEAVE



~150 micron

DISPLACEMENT VECTOR MAG: 10X

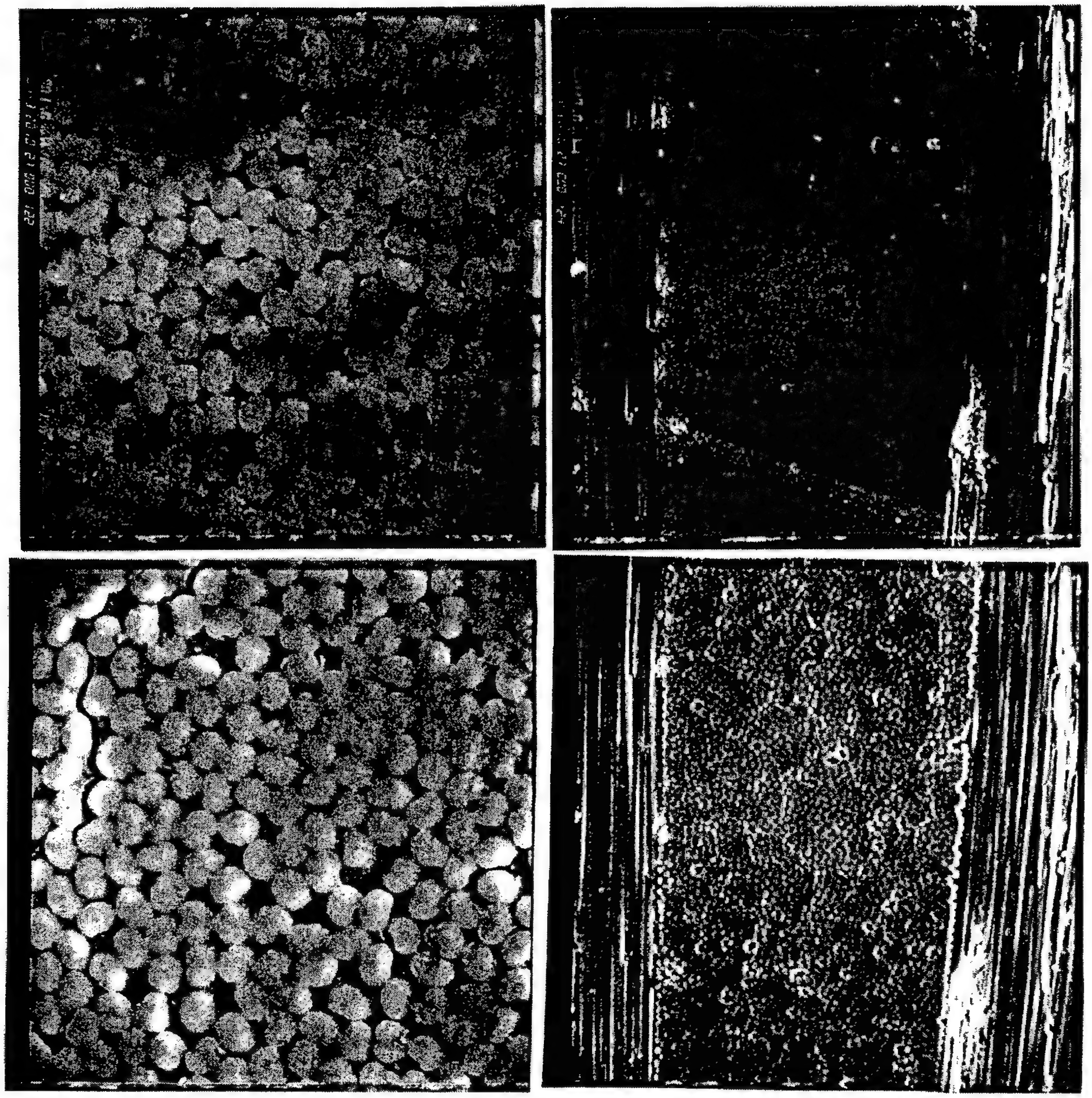
CENTRAL REGION OF LEFT PHOTO



~30 micron

DISPLACEMENT VECTOR MAG: 6X

[20] A. Nagy, J B Campbell, and D L Davidson, Rev. Sci. Instrum. 55(5). May 1984.
[21] D L Davidson, K S Chan, and R A Page, AMD-Vol. 102, Micromechanics: Experimental Techniques (Ed: W N Shapiro Jr) ASME Book No H00539 - 1989.
[22] Note that the photos under the displacement maps are rotated 90 degrees and flipped with respect to the photos on the next page; the regions viewed are the same.



High Magnification

Low Magnification

Top row: 23 C, as cured

Bottom: at 590 C, mostly carbonized

Photos from hot stage in SEM at SWRI. Sample T40R-0.4, a cloth-reinforced phenolic-matrix laminate as-cured at room temperature and as-carbonized in the microscope at 590 C. Scales of magnification indicated on displacement maps on previous page.

3.0 IMAGE ANALYSES

Three measures of fiber array geometry have been explored with the aim of providing a quantification that is relevant to shrinkage mechanics. These three measures are:

a. fiber contact density
b. fiber contact angle
c. intra-matrix distance statistics

3.1 Fiber Contact Statistics

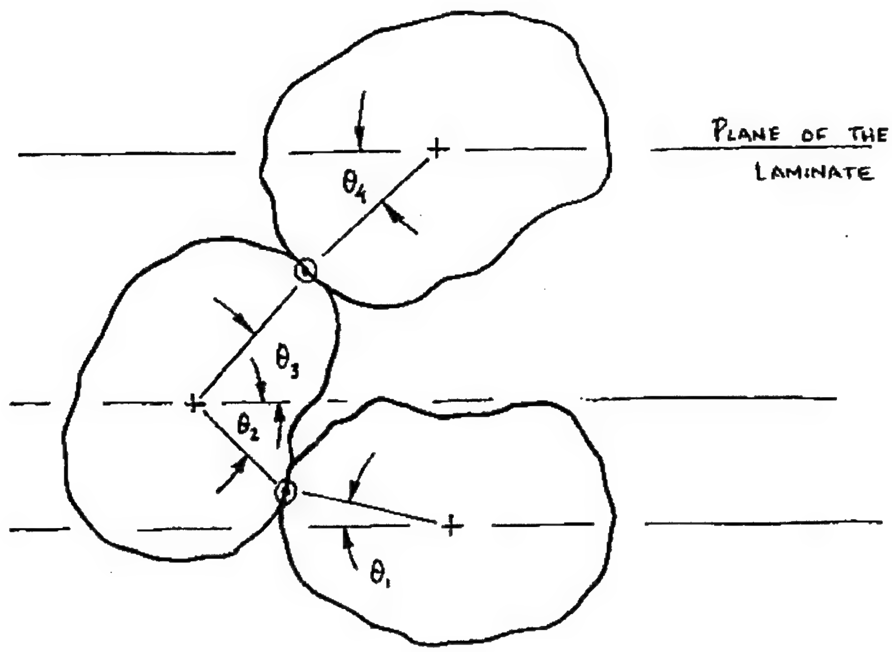
To help quantify the spatial distribution of fibers within tows, two measures have been defined. *Fiber contact density* is the number of fiber-fiber contacts per fiber cross-section. For example, a fiber cross-section in contact with two neighboring cross-sections has a fiber contact density of two. *Fiber contact angle* is the angle between the plane of the laminate and the line joining the centroid of a fiber cross-section to the center point of the contact with a neighboring section (see sketch on next page). A semi-automated computer-aided procedure is applied to digitized SEM images that show fiber cross-sections. The sequence of image manipulations is shown on the following pages. An example of the results is shown in Figures 3.1-1 and 3.1-2. We see that fiber contact density distributions are quite different for the two as-molded composites in Fig 3.1-1, a result that encourages the use of the method in our further work.

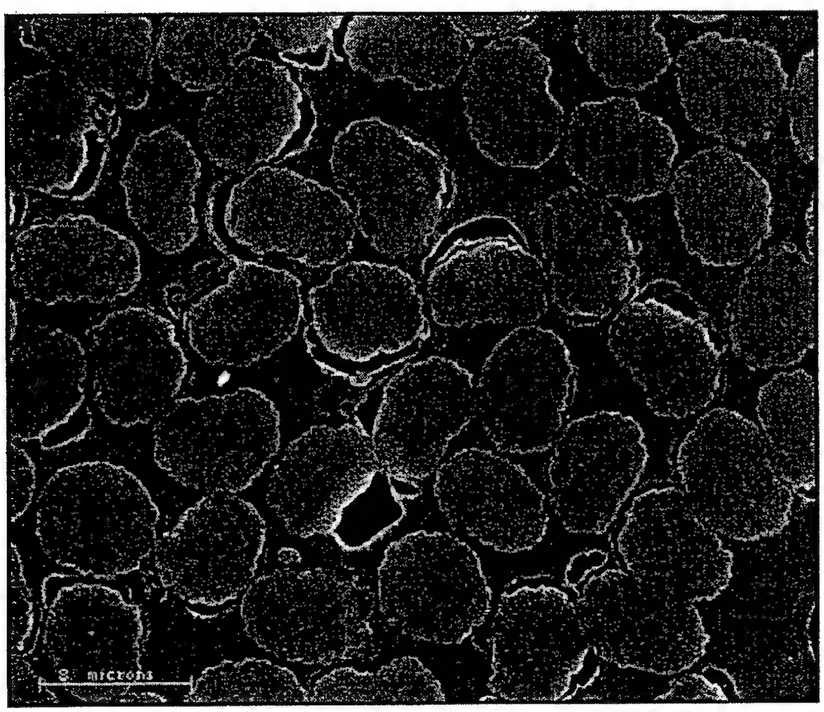
Since the fiber array geometry will, on average, be the same on any cross-section through a tow, the fiber contact density distribution gives an estimate of the fraction of fiber length that is unsupported by neighboring fibers. For example, from the frequency of zero-contact fibers, Material A in Fig. 3-1 has about 33% of the fiber length unsupported; Material B has virtually no unsupported length. The unsupported length is expected to be important to estimating the elastic deformations of the fibrous reinforcements, via bending and/or contact-deformation models.

Samples of the composites represented in Fig.3.1-2 also were examined in the as-cured (molded) condition. The fiber volume fractions of these samples, estimated by point-count and area-fraction analyses of SEM images, were unusually high (approximately 0.78 to 0.80). Thus, the potential effects of surface treatment on fiber contact density are much attenuated, because such high volume fractions tend to full packing. Nevertheless, as Fig. 3.1-2 shows, there is a tendency for the less treated fibers to have higher contact densities, in keeping with the hypothesis advanced earlier.

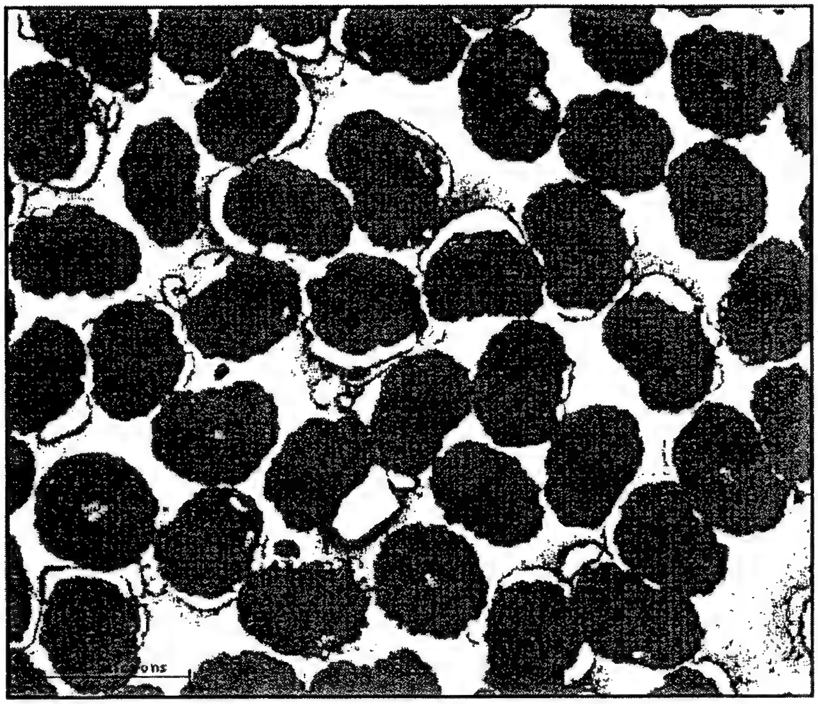
Additional statistics were gathered from polished samples of composites listed in Tables A and B. The results, summarized in Figures 3.1-3 through 3.1-5, tend to support the hypothesis that less wetting leads to more clustering and contacts and therefore to lesser shrinkage in carbonization. However, more work would be needed to give statistical significance to these inferences.

Definition of contact angles and sequence of images used for fiber contact analyses...

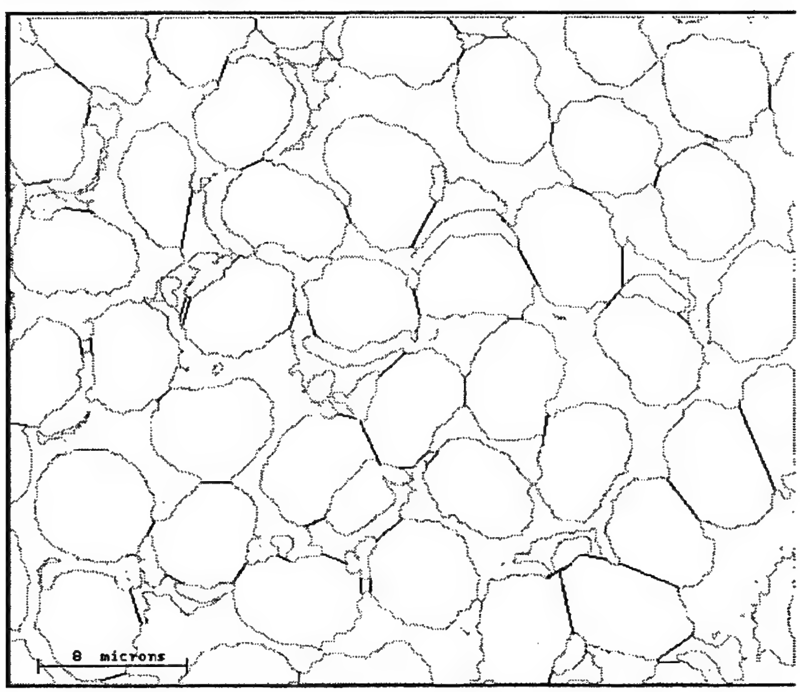




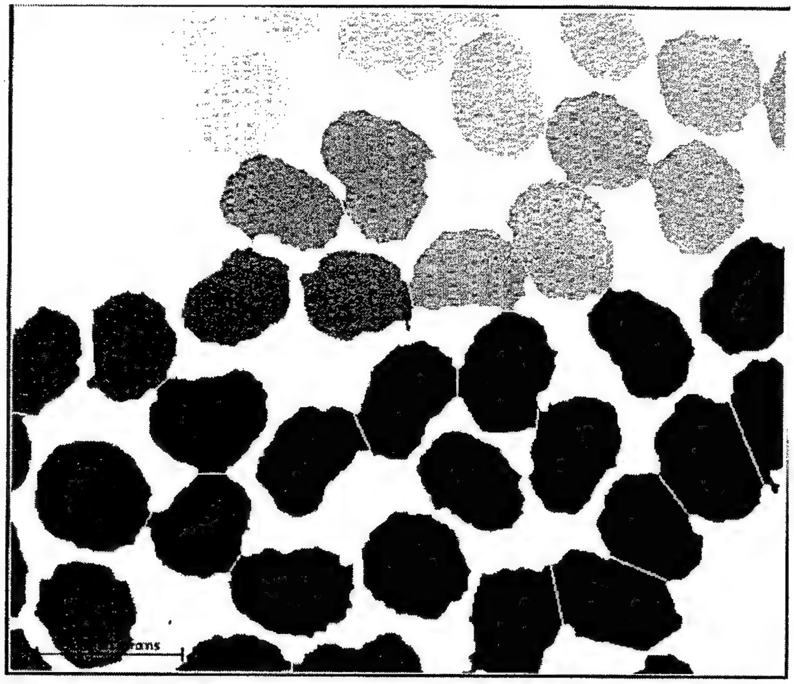
Original SEM digital image, CCAT material.



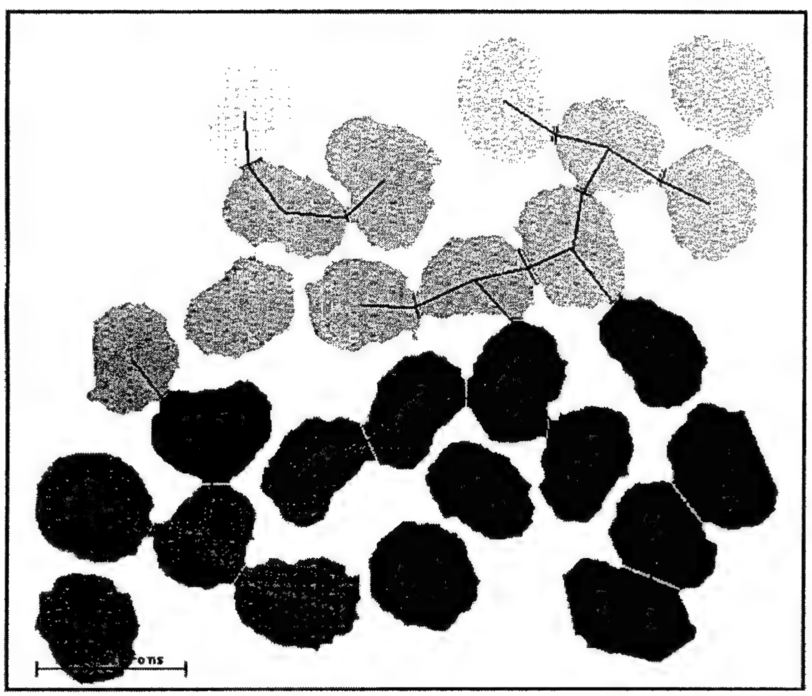
Thresholded image from Figure 1.



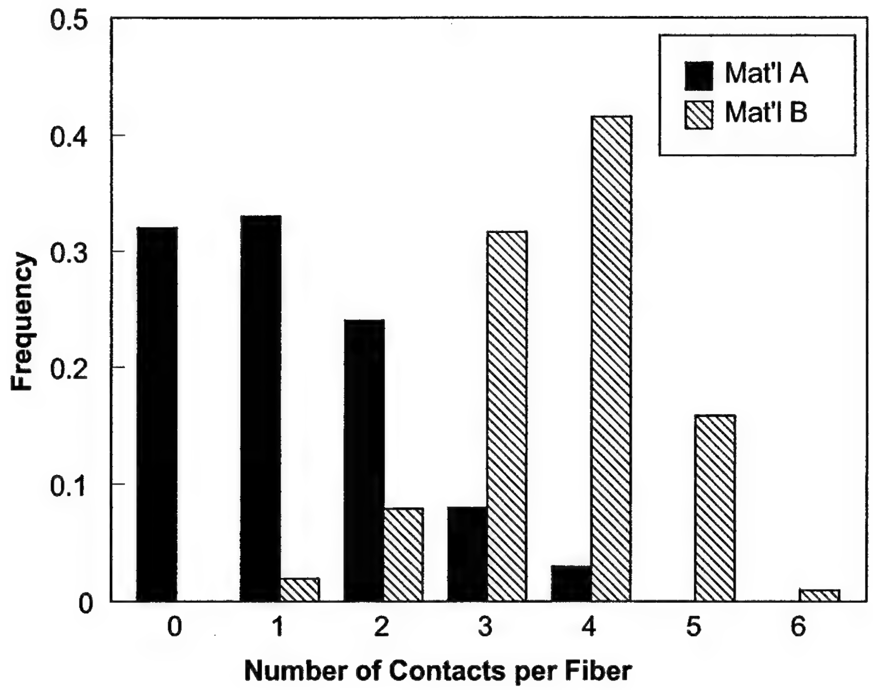
Fiber edges are detected. Contacting fibers are manually segmented with line segments across the necks. Fill seed points are also manually inserted into the central region of each fiber.

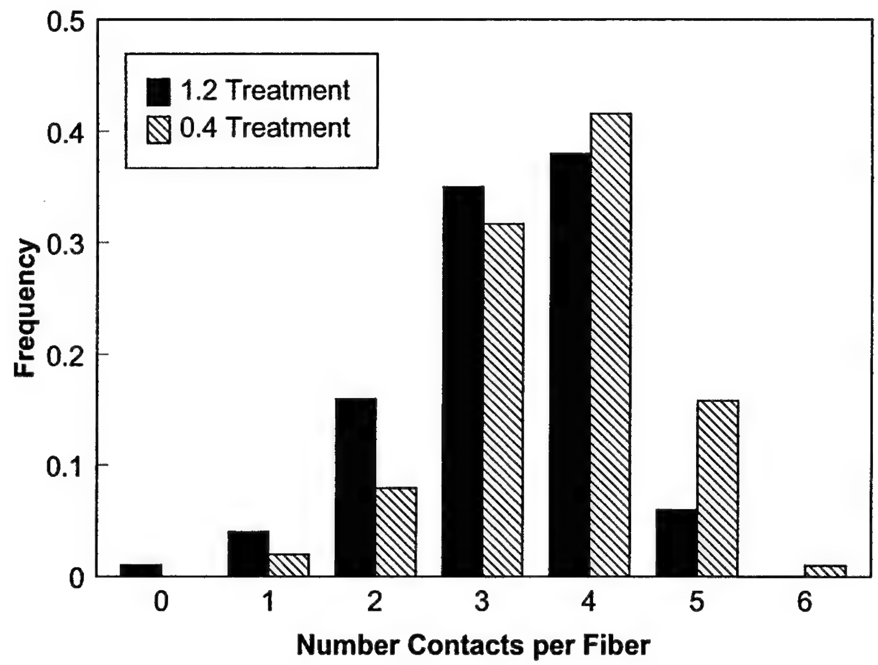


Fibers are filled and identified as individuals (labeled as colors). All other background features in the image are discarded.



Contact density and contact angles are measured. Edge fibers are excluded. Contact angle vectors are drawn to provide visual confirmation that the code is working properly.





Upper chart: **Fig. 3.1-1**. Fiber contact densities of two composites. Material A: T300/phenolic, with .58 fiber volume fraction; Material B: T40R/ph, with .79 fiber volume fraction.

Lower chart: **Fig. 3.1-2**. Fiber contact density distributions for two T40R/phenolic laminates of different surface treatment levels (See Table B)

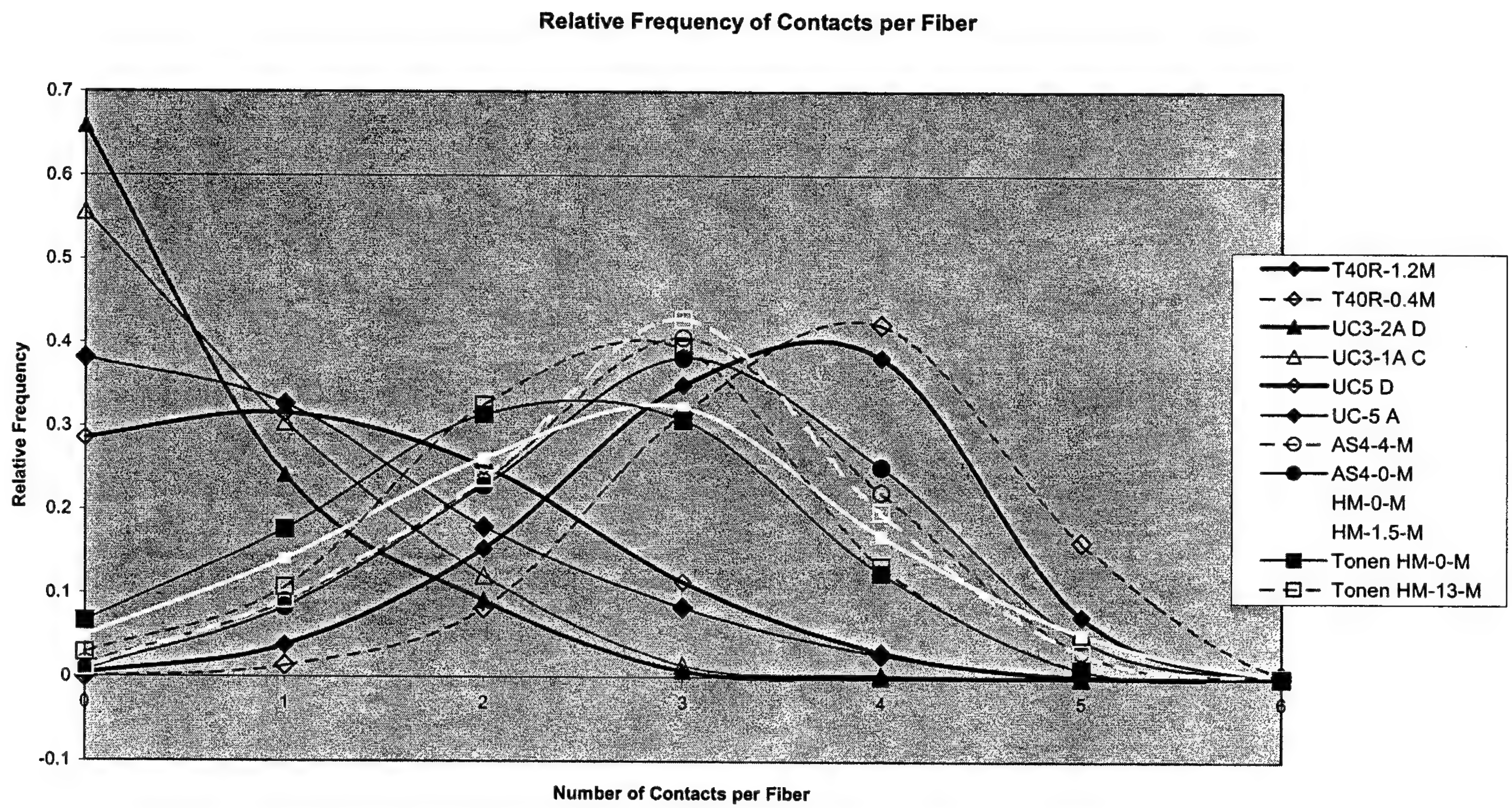


Fig. 3.1-3. Histogram of contact density (contacts per fiber cross-section), as measured on micrographs of various samples of carbon/phenolic composite (Tables A and B). The lines joining the points have no mathematical significance, but are provided only for ease of seeing trends.

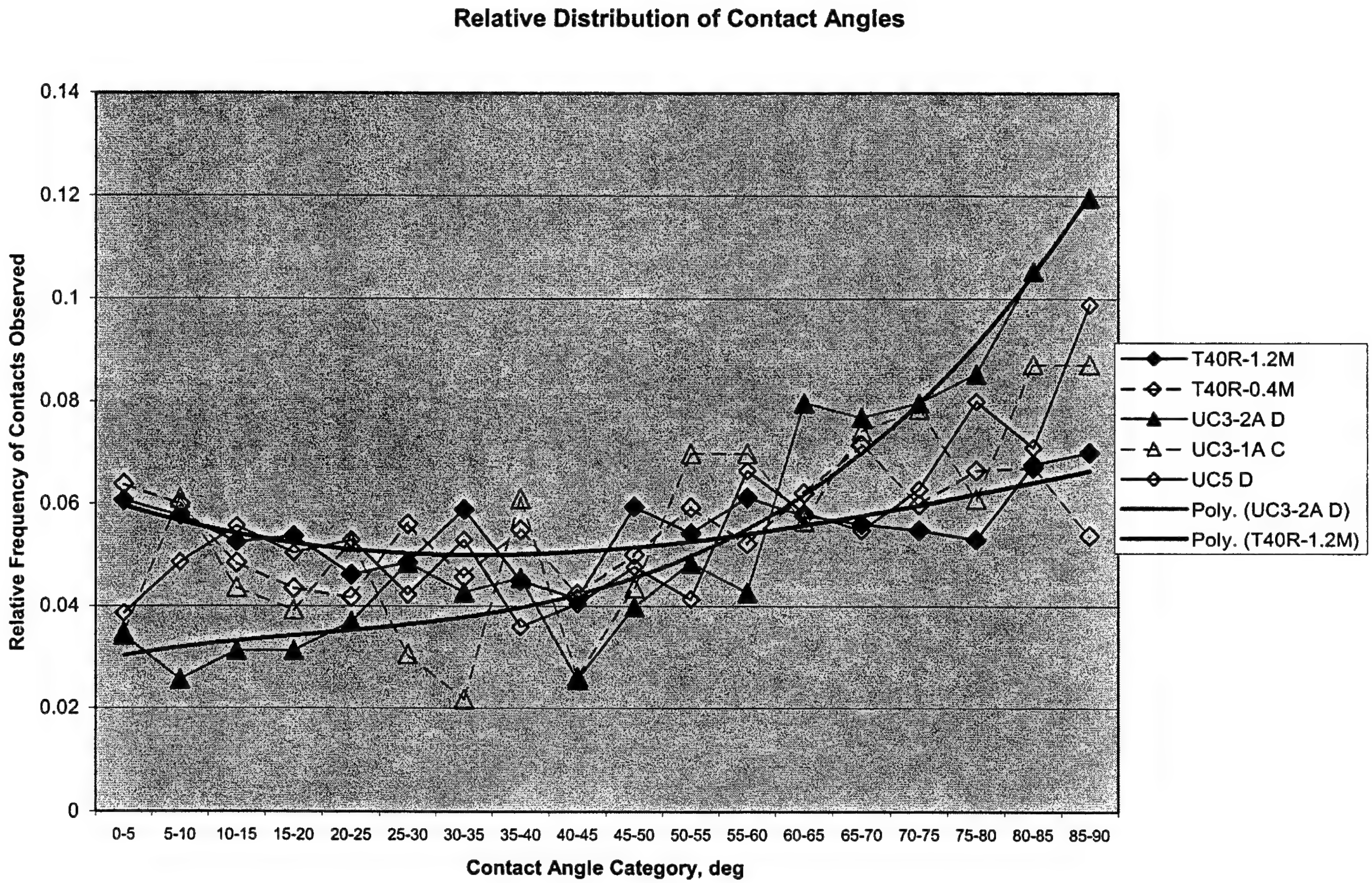


Fig 3.1-4. Histogram of contact angles for selected materials shown in Tables A and B. For samples of low fiber volume fraction (UCx) there is a clear trend towards more contacts in the direction of compaction. For samples of very high volume fraction (T40R-xx) there appears to be little trend with direction.

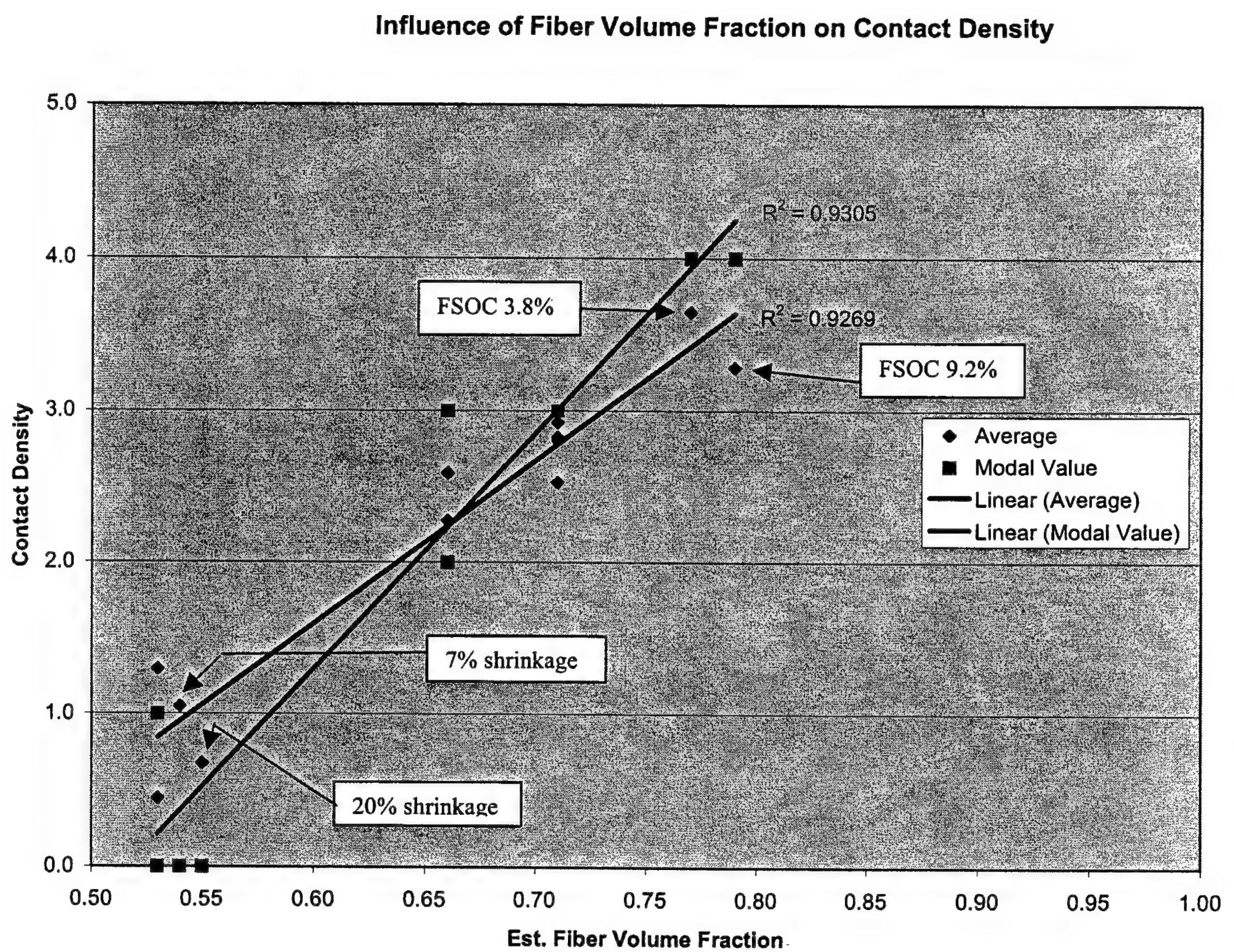


Fig. 3.1-5. Variation of contact density with fiber volume fraction, for materials in Tables A and B. The general trend is with fiber volume fraction, but at a given volume fraction there is secondary effect such that higher contact densities correspond to lesser shrinkage in carbonization (Table B) and lesser wetting (as characterized by fiber-surface oxygen content).

3.2 Intra-Matrix Distance Statistics

Another approach to characterizing fiber arrays has been explored. The idea is that the distribution of fiber-to-fiber distances, between a fiber and its nearest neighbors, should reflect the degree to which the array is clustered or well separated. Several small BASIC programs were written to play with this concept (see Appendix 2).

The first attempts dealt with arrays of perfect circles of equal diameter. Random arrays were constructed, and also several types of well-defined arrays for any given area fraction of circles. The well defined arrays include close-packed regular hexagonal and close-packed contacting hexagonal with fibers randomly removed to achieve the desired area fraction. Line lengths in matrix are defined as in Fig. 3.2-1.

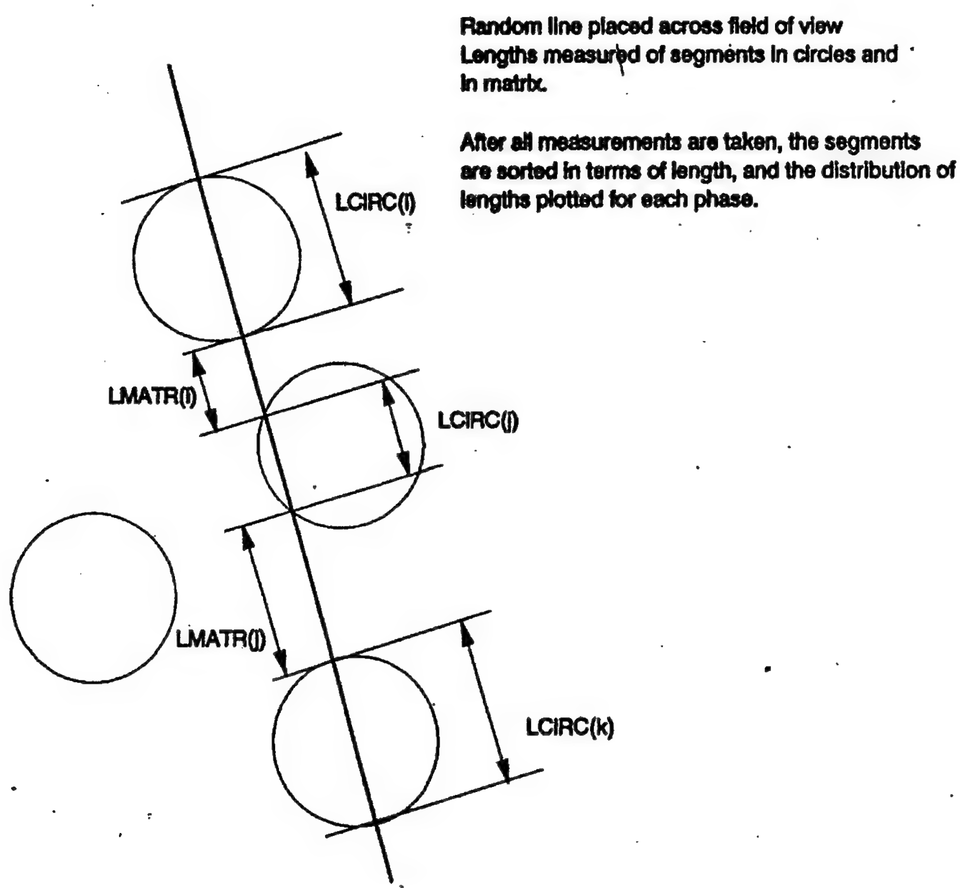


Fig. 3.2-1 Definition of intra-matrix line-segment lengths.

In this work, these lengths were compared to two specific lengths of matrix in a regular close-packed hexagonal array of the same area fraction. The two specific lengths of the hex array are the center-to-center distance, and the minimum gap between circles (referred to as the proximity distance).

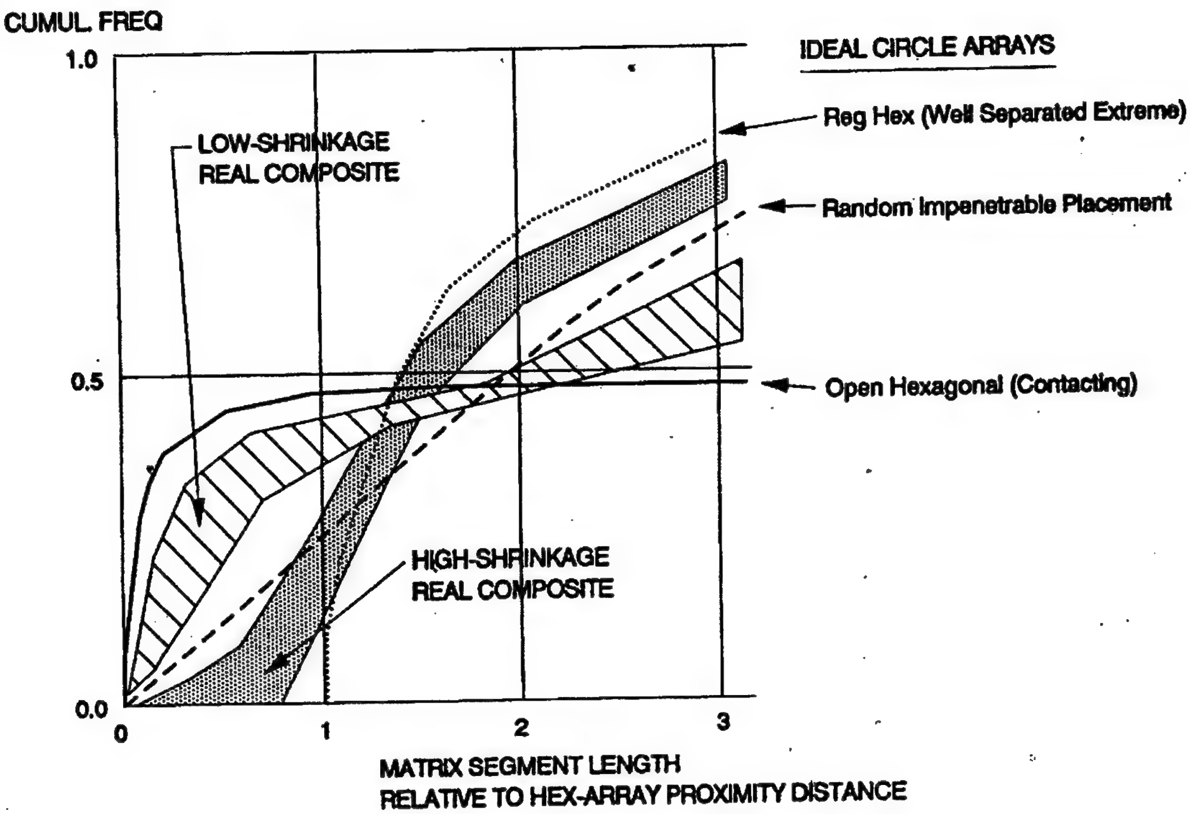


Fig. 3.2-2 Cartoon showing potential differences in intra-matrix line-length statistics; lengths normalized by nearest neighbor gap distance in regular hexagonal array of same area fraction.

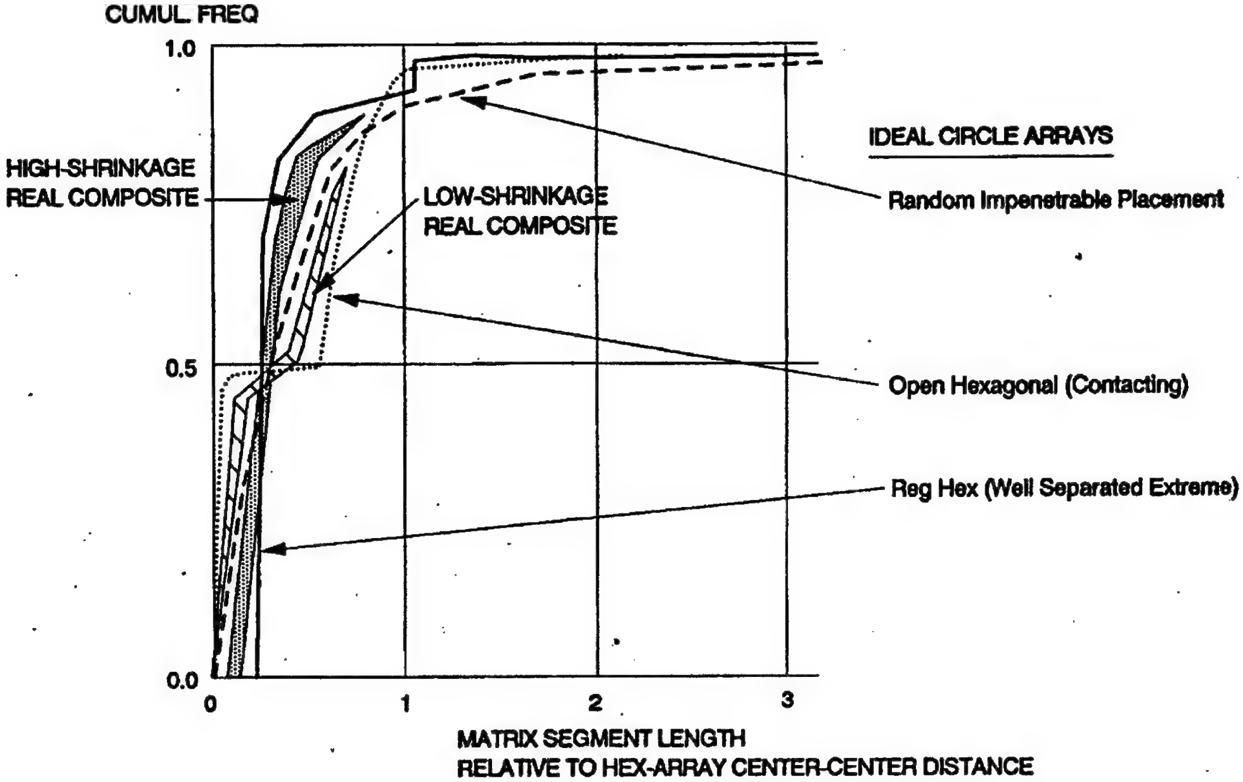


Fig. 3.2-3. Cartoon showing potential differences in intra-matrix line-length statistics; lengths normalized by nearest neighbor gap distance in regular hexagonal array of same area fraction.

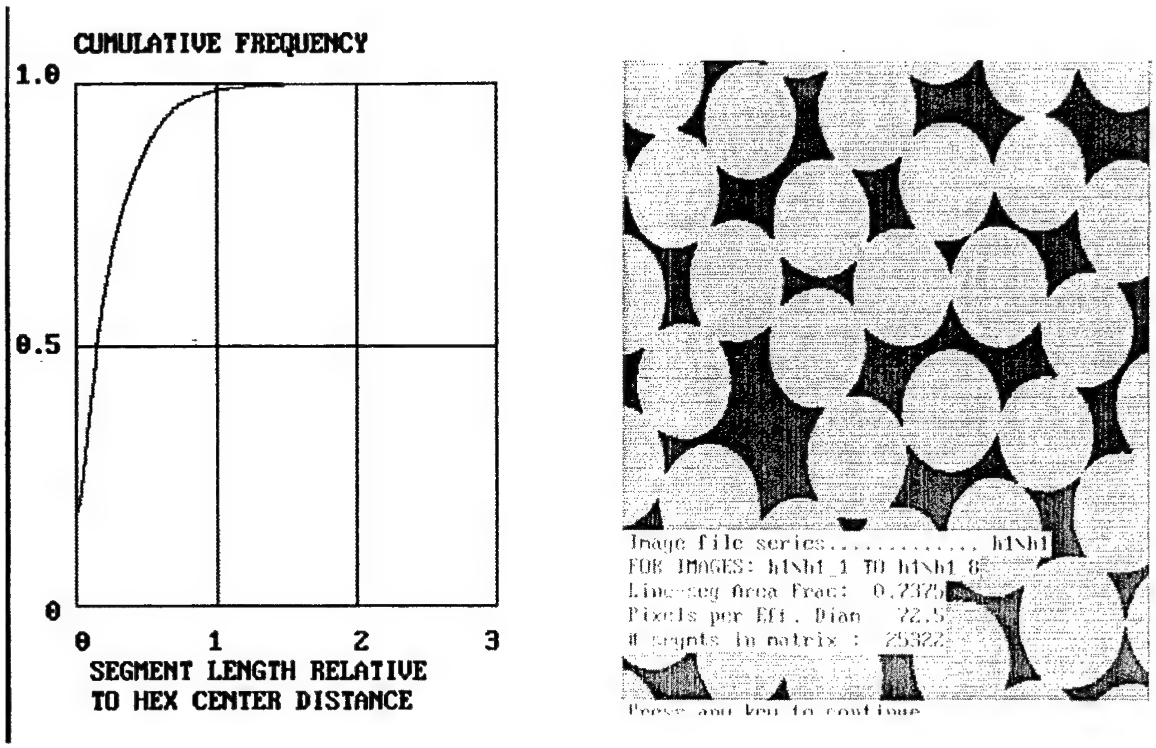
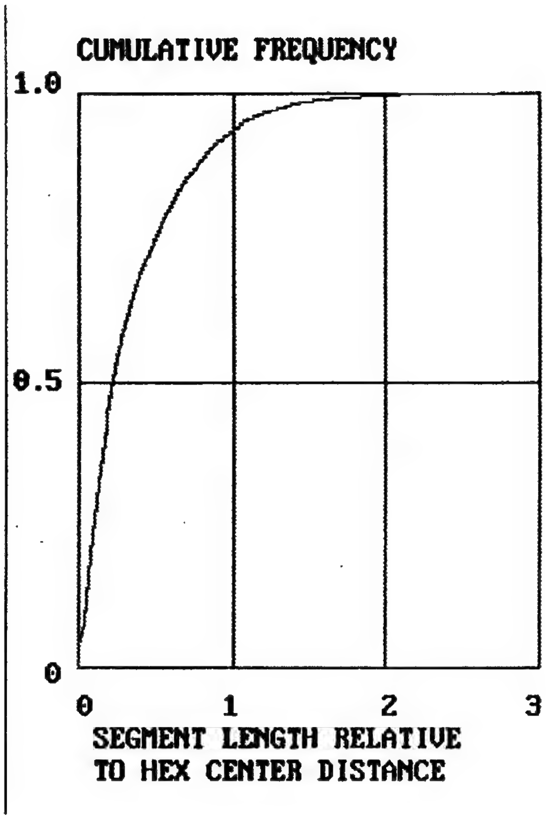


Fig 3.2-4. Line statistics re: center-center distance for regular hex array of circles of same area fraction, composite H1 Statistics from about 7 to 12 images of type shown on right.



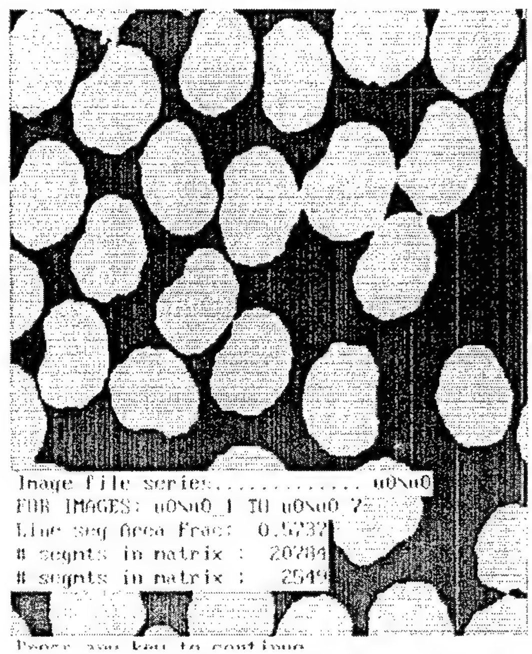


Fig 3.2-5. Line statistics re: center-center distance for regular hex array of circles of same area fraction, composite U0 Statistics from about 7 to 12 images of type shown on right.

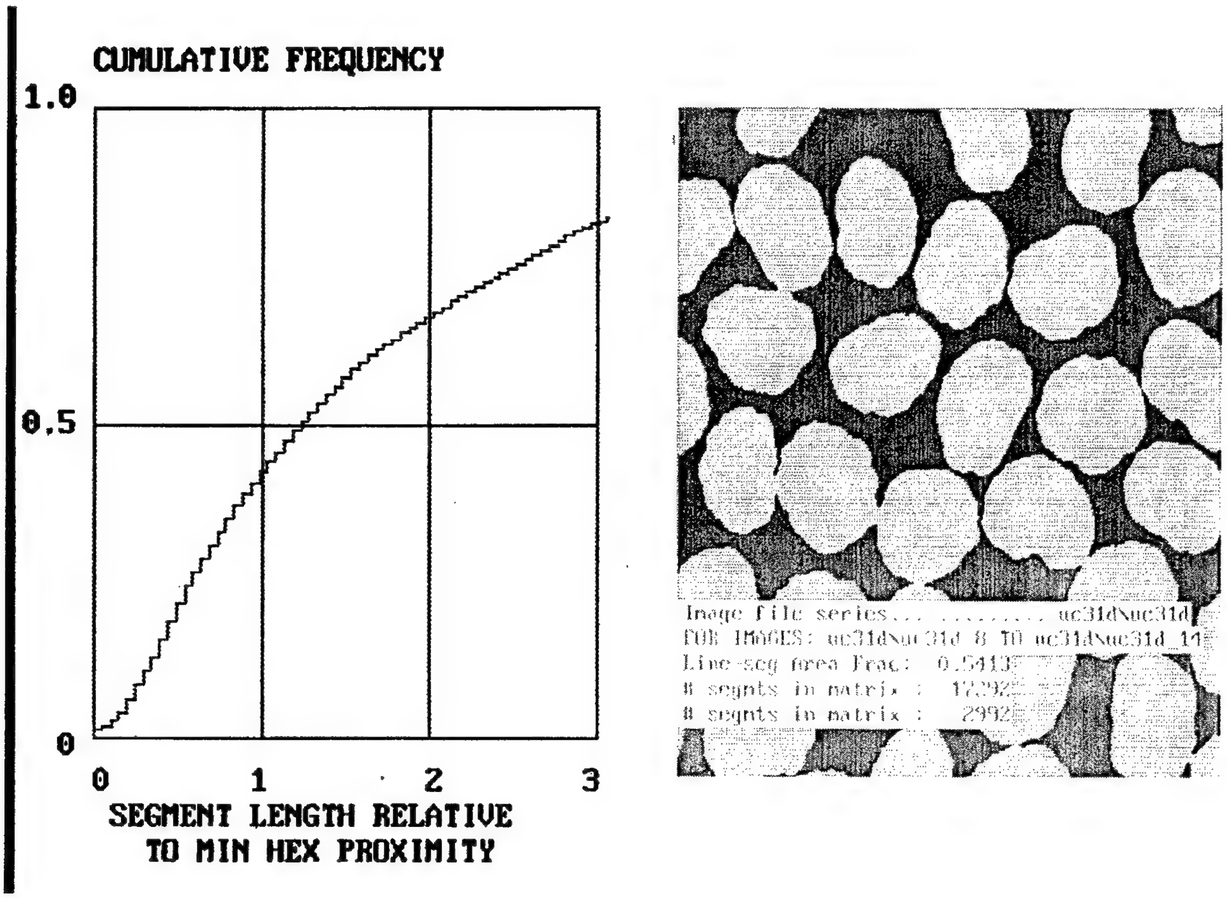


Fig. 3.2-6. Line statistics re: minimum proximity distance in regular hex array of same area fraction. Composite UC31D. Statistics from about 7 to 12 images of type shown on right.

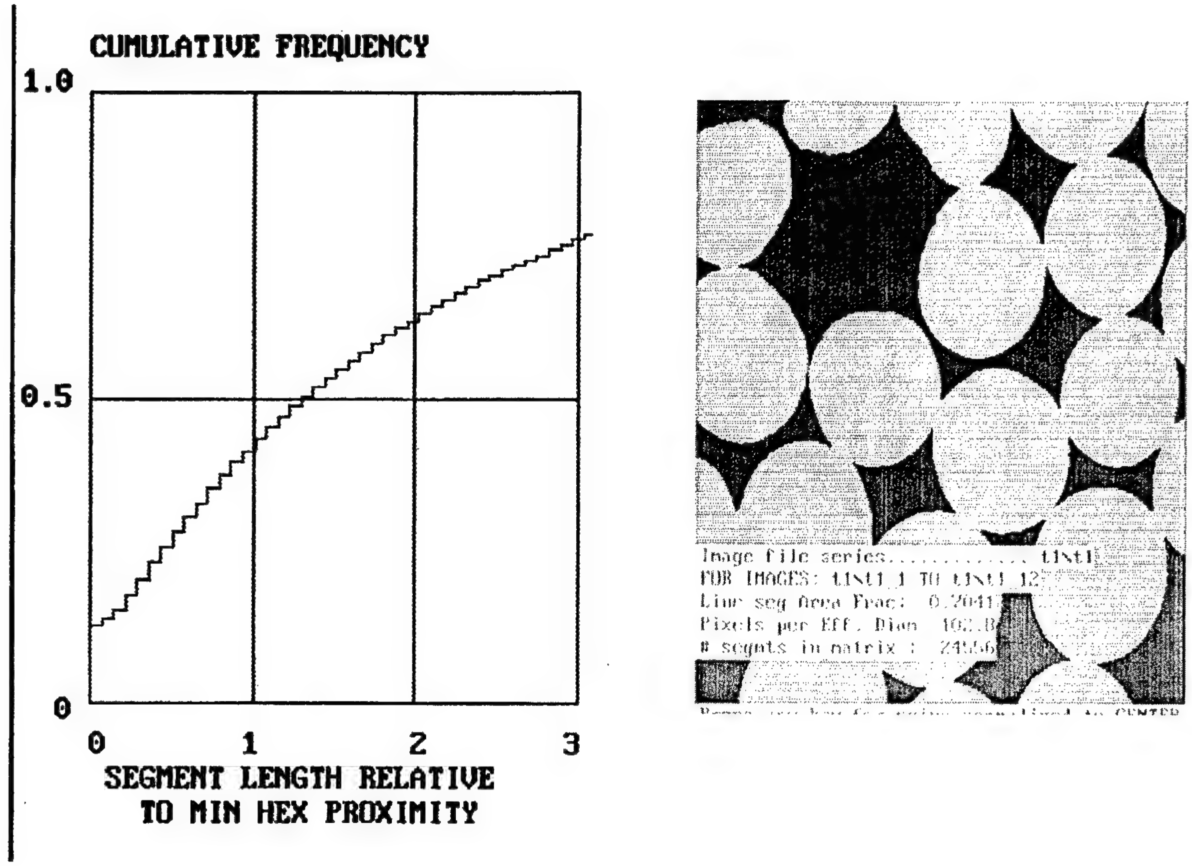


Fig. 3.2-7. Line statistics re: minimum proximity distance in regular hex array of same area fraction. Composite T1. Statistics from about 7 to 12 images of type shown on right.

The statistical ogives for real microstructures, like those shown above, do not fully conform with preconceptions (Fig. 3.2-2 and 3.2-3). However, there are enough significant differences in line statistics due to fiber volume fraction and array type (clustered or separated) to suggest that further studies of this sort would be worthwhile.

Table A. Description of Acurex Samples for Microstructural Image Analysis Available Data [1]

Code (a)	Approx Size mm (b)	Fiber (c)	Matrix (d)	Approx Avg Fiber Volume Fraction (Cured)	Thickness Shrinkage percent on Heating to 815 C
UC-3-1A	8x24x10	T-50	phenolic	0.50	20
UC-3-2A	7x15x12	T-50	phenolic	0.61	10[23]
UC-5A	7x24x10	T-300 (HT)	phenolic	0.54	7

Notes

a. These unidirectional composite samples were manufactured by Acurex Corp, under the supervision of Knight Leong and Jim Zimmer, ca 1985-86[24]. The letter A, at the end of the code, means these samples are in the as-cured condition.

b. The last dimension is along the fibers. One face, perpendicular to the fiber axes, is partly polished as received from Acurex. The circular-saw cut(easily seen opposite to the polished face) was done with a 4"-dia diamond-grit blade with water cooling, after which the sample was patted dry with a paper towel and allowed to air dry overnight (9 Feb 94, Orange County, CA) before being placed in the plastic bag for shipping.

c. Thornel 300 and Thornel 50 carbon fibers are said to be from the same PAN precursor; the processes by which the PAN is converted to carbon differ at least as regards the peak heat-treatment temperature, which is higher for T-50 than for T-300. The T-300 used here, designated HT, has been deliberately heated to above 2000 C, in an inert atmosphere, before use in the composite; this would have removed any surface sizing. The T-50 fiber used here was un-sized.

d. The phenolic resin is known as Karbon 640 (or K640). Cured at 175 C.

---

[23] Compaction of sample UC-3-2A is distinctly uneven, with one half the thickness at much higher fiber volume fraction than the other half. We do not know why the stratification. We will ignore the significance of the shrinkage number, which incorporates both regions.

[24] K. Leong, J. Zimmer, and R. Weitz, Fiber Property Changes During Processing of Carbon-Carbon Composites, AFWAL-TR-87-4035, June 1987.

Table B. Description of NASA Langley Samples for Microstructural Image Analysis

Specimen	Description	Surf Trtmt Level	FSOC atom %	Thickness, in As cured	Thickness, in Carbonized	Growth	Approx Fib Vol %
T40R-0.4M	PAN based carbon fabric	0.4	3.8	0.0939	0.0956	0.0017	79
T40R-1.2M	"	1.2	9.2	0.0973	0.0944	-0.0029	77
AS4-0-M	PAN based carbon 1D	0.0	2.2	0.0943	0.0958	0.0015	71
AS4-4-M	"	4.0	11.9	0.0990	0.0967	-0.0023	71
HM-0-M	PAN-based carbon 1D	0.0	1.6	0.0915	0.0943	0.0028	66
HM-1.5-M	"	1.5	7.2	0.1040	0.1027	-0.0013	71
Tonen HM-0-M	pitch based carbon 1D	0.0	2.0	0.0972	0.1006	0.0034	66
Tonen HM-13-M	"	13.0	5.0	0.0910	0.0922	0.0012	71

Notes: Surf Trtmt Level = fiber surface treatment level relative to manufacturer's standard commercial treatment
FSOC = atom % of oxygen at fiber surface measured by XPS
All samples in as-cured process state (phenolic resin)
Thickness as measured by NASA, an average of panel measurements
Approx Fib Vol % = intra bundle fiber volume percent estimated from image analyses at Clarkson

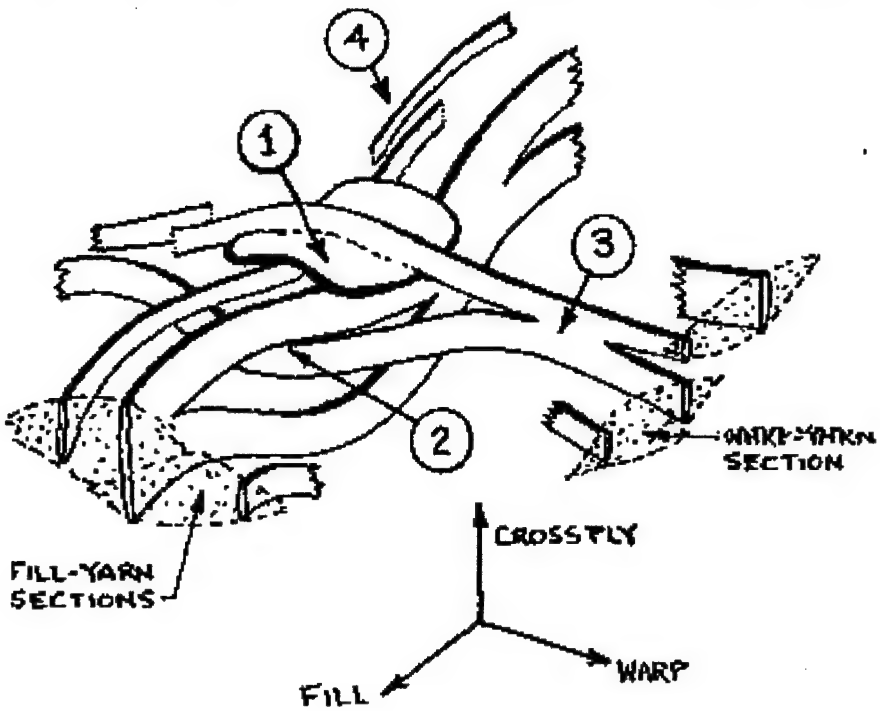
4.0 THREE-DIMENSIONAL EFFECTS

The discussion so far has been mainly in two-dimensional terms, applying to the clustering of fibers within uncracked areas of yarn cross-sections. This section takes note of some essential three-dimensional aspects of the real problem.

4.1 Crack Networks in a Plain-Weave C/C Laminate

The sketch below is a schematic illustrating selected features of microcrack channels in a plain-weave carbon-carbon laminate:

1. yarn interface crack joining orthogonal sets of transverse bundle cracks
2. "point" junction of orthogonal transverse bundle cracks
3. merging and splitting of transverse bundle cracks in two parallel yarns tapering off and overlap of transverse bundle cracks within one yarn bundle
4. tapering off and overlap of transverse bundle cracks within one yarn bundle

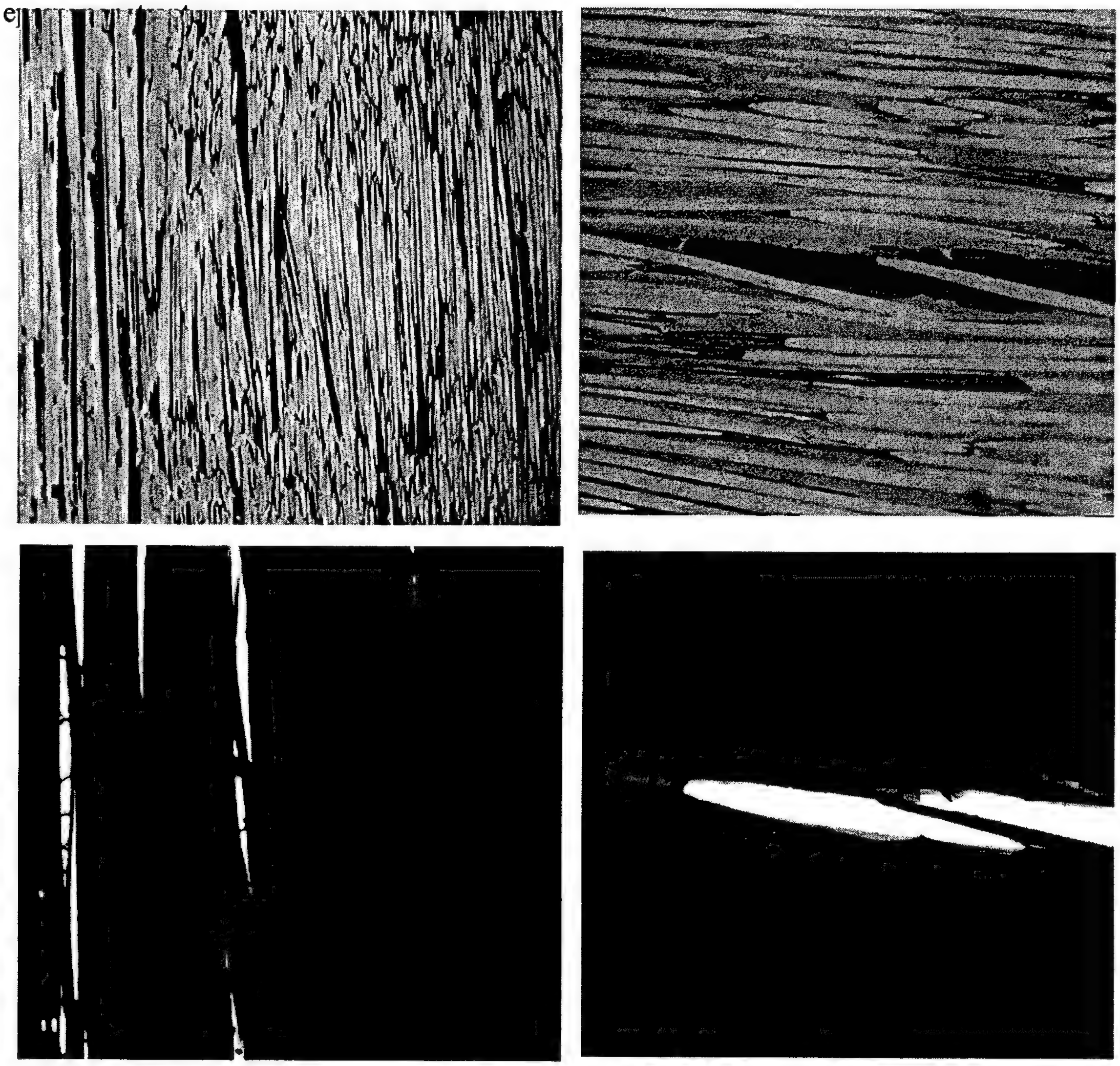


Sketch: Schematic of crack geometry in densified and graphitized plain-weave carbon-carbon laminate (by Jortner, based on sequential sectioning results produced by Yurgartis and Mulvaney at Clarkson University, April 1993) (The indistinct lettering at the right side reads "WARP-YARN SECTION")

4.2 Fiber Bridges Across Yarn-Bundle Cracks

The previous sketch is observed at the level of yarn bundles. On closer examination, at the fiber level, as in photomicrographs below of another carbon-carbon laminate, we see that occasional fibers cross transverse bundle cracks. It is difficult to estimate how often this happens, but such crossings must be common, otherwise we would not have captured so many examples. Why do crossings occur? Perhaps it is the twist in the tows. But it does not appear that these tows have much twist over the area of these photos. Perhaps there is local commingling of the fibers (due to deviations from fiber straightness, or waviness) that causes the growth of a crack to switch fiber sides. There is also the fiber bowing in the interlace regions, so a crack that is trying to grow normal to one stress will have to occasionally index across fibers.

Photos below: Two views of yarns in cloth reinforced carbon-carbon laminate, in densified condition, looking down on the plane of the laminate. Upper photos are normal optical micrographs. Lower photos show dark field views of same areas impregnated with fluorescing e



4.3 Visualization of Fiber Paths Along Yarn Axis

Yurgartis and Guski at Clarkson U have achieved the capability of serial sectioning composites at the fiber-fiber level of detail, so that the relation of a single fiber to its neighboring fibers can be traced along the length of a fiber bundle. Initial serial photos (viewed as a digitized movie on a computer)[25] show some interesting features: for example, in the samples examined, individual fibers wander from one contact cluster to another along the bundle length. This additional degree of freedom is bound to influence our models of composite compaction stresses and stiffness of contact arrays; in essence, the discovery suggests that 2D models must be viewed with caution unless some allowance is made for lengthwise variations of contact structure. There also is a hint of helical rotation in the paths of small fiber clusters.

Yurgartis reported informally,

> The movies themselves are all of the UC-5 material. One fiber in the first frame of each movie has a white "+" on it. This is the fiber that was used as the reference point. Imagine standing on this fiber and viewing the pictures. The jumping of the image frames is due to the limitations of the positioning reproducibility. (We need to fix this. We have two plans to improve the situation: a modification of the specimen holder that eliminates the use of potting epoxy as a location controller, and a controlled translation microscope stage that will let us center on a particular fiber.)
>
> Three different positions on the sample were studied. The first section, however, is broken into two movies since no fiber remained in the viewing area through the entire sectioning process. The red bar on the right of the frame will be seen to shrink as the sections progress. The bar represents the section depth. Full scale is 500 microns. The total section depth of movies 2 & 3 was around 580 microns, so the bar vanishes during the last 8 or 9 frames.
>
> Something interesting that I see in the movies is the appearance of groups of fibers, about 4-10, that form loose rotating clusters. The rotation of the clusters is particularly interesting. Why the small clusters? Something in the way the tows are constructed? Why the rotation? It is not gross tow twist, but is there a twisted substructure in the tows?

[25] The digitized movies are viewable with an mpeg player appropriate to the computer being used. Files for four movies obtained via the serial sectioning technique are available from Jortner via diskette or email (jjortner@oregoncoast.com).

5.0 SCATTER OF ENGINEERING PROPERTIES OF CARBON-CARBONS

Scatter of mechanical properties can be fairly large for many carbon-carbon composites. Observed coefficients of variation (cv) for strengths and stiffnesses typically are in the range of 10% to 15%, sometimes as low as 7-8% and sometimes as high as 25% or so. In contrast, common aluminum alloys have cv of only 3-5% for their yield strengths. Therefore, to attain the same probability of survival under load, carbon-carbon structures must be designed to higher margins of safety than aluminum structures, not only because of the difference in cv, but also because the number of test replications usually is smaller for the carbon-carbons. If we compare a metal with 5% cv of strength measured on 100 replicate tests to a composite with 10% cv based on 10 replications, and we require 99% reliability with 95% confidence, we find (for normally distributed data) that the peak design stress for the metal can be about 85% of its mean (yield) strength, whereas the composite can be stressed only to about 40% of its mean strength. Such dismal statistics serve to focus attention on the causes of property scatter in carbon-carbons, with hopes of finding ways to reduce variability.

The cv get larger, in a broad sense, as we go from metals to plastic-matrix composites to carbon-matrix composites; this progression corresponds to increasing microgeometrical complexities and difficulties of manufacture. Surprising to this writer is the degree of overlap. For example, the cv for strengths of some metals (including some 300-series stainless steels and beryllium alloys) range up to about 15%; cv reported for cloth-reinforced carbon/epoxy laminates cover the range between 2% and 17% or so, with most data between 10% and 15%. Thus, these other well accepted structural materials exhibit about the same scatter as many carbon-carbons (although, with less test replications, confidence in the carbon-carbon cv typically is not as great at present). With regard to the many carbon-carbons made from resin-matrix laminates, we may ask, "Why is the cv for carbon-carbon no greater than the cv for resin-matrix composites, in view of the difficult processes used to convert a carbon-phenolic, say, to a carbon-carbon?"

There is some evidence that densification processes reduce variability. Tensile strengths presented by Naughton and Roy in 1993[26] can be analyzed to show that samples grouped into four disparate batches after the initial carbonization of the cured resin-matrix laminate become, statistically speaking, members of a single batch after two densification cycles. This inference is based on the F-test, which compares cv calculated assuming each batch is from a different population to the cv calculated assuming all samples belong to one batch. Thus, while the carbon-carbon processing introduces new sources of variability (like shrinkage during first carbonization of some matrices, discussed below), densification itself may tend to reduce overall scatter.

Carbon-precursor resins undergo considerable shrinkages during carbonization. Phenolic resins, for example, shrink about 50% in volume from normal cure to 1000 C. Some resins shrink less, but the minimum seems to be about 20-30% in volume. The composite, however, may or may not shrink in response to the matrix shrinkage, depending on the fiber volume fraction, the reinforcement geometry, and some details of the fiber array within fiber bundles. Data show composite shrinkage decreases with increasing volume fraction of fibers. At a high enough

[26] See Naughton and Roy, Extended Abstracts of the Biennial Conf on Carbon, American Carbon Society, 1993. We rely also on privately transmitted data from Ajit Roy.

volume fraction, composite shrinkage can be zero; above that fraction, expansion rather shrinkage can occur. Except at the zero-shrinkage level, fiber volume fraction is changed by carbonization. When filaments within bundles are well separated by matrix, composite shrinkage tends to be larger than when, at the same fiber volume fraction, there is a high density of filament-filament contacts. If the composite shrinks less than the volume lost by its matrix, the deficit is associated with porosity. In addition to the material characteristics, the geometry of the structure (curved or not, for example) may affect the composite shrinkage. At the least, shrinkage variations affect properties that depend on porosity and fiber volume fractions.

Increasing anisotropy of the composite as it becomes a carbon-carbon also contributes to scatter. Examples include increased sensitivity of properties to local deviations in fiber orientation (as with crimp in woven reinforcements), increased possibility of nonstandard failure modes (like yarn pullouts and splintering at surfaces), and increased sensitivity to variations in loading direction. There seem to be opportunities for optimization of weaves to minimize scatter. An effect related to anisotropy is the bimodular change in stiffnesses caused by prior loading in certain off-yarn directions.

Property scatter is seen as a major limitation to the wider applicability of carbon-carbons, in both military and civilian uses. Among the problems is the fact that the composite part used in an application has complex shape and varying architecture, whereas the specimens tested to provide design properties are commonly excised from specially made simple shapes (flat panels or circular cylinders). Insofar as geometric restraint influences shrinkage, properties measured on samples from panels or cylinders should be adjusted to apply properly to shaped parts. An approach to making such adjustments would improve reliability of carbon-carbon structures.

Several paths towards reducing scatter and its effects appear worth mentioning.. Preliminary prescriptions include:

a. Make the composite with fiber volume fractions high enough to produce zero composite shrinkage on carbonization.
b. For composites that must be made with low fiber volume fractions, give careful attention to the details that control the degree of filament-filament contact within fiber bundles, including fiber-matrix wetting and the temperature and pressure histories during compaction and cure.
c. Select heat treatment temperatures high enough to reach the density plateau for the matrix in question.
d. Increase the multidirectionality of the reinforcement and pay attention to the loads imposed on the material prior to service or testing, to minimize load-history variabilities and their effects on stress-strain responses.
e. Plan the approach to property testing to include analytical or experimental evaluation of possible differences between properties in test articles made from simple shapes and the properties in more complex structures.
f. Conduct as many replications of property tests as economically possible.

These, of course, are merely suggestions for potentially beneficial approaches, without guarantee as to appropriateness or efficacy under any particular circumstances.

6.0 CONCLUDING REMARKS

In this work, we have focused on the factors that affect transverse shrinkage during carbonization of fiber bundles in composites of carbon fibers in a thermoset resin matrix. Our results add to the evidence that fiber arrays in the cured-resin state affect the shrinkage during carbonization.

Composite shrinkage during carbonization is an example of a composite behavior that is affected significantly by details of fiber-array geometry (not just the fiber-volume fraction). It is true that fiber volume fraction is a major influence. However, shrinkage also is affected significantly by the geometry of the fiber array - specifically the extent to which fibers are in contact with each other thereby restraining compressive deformations. The effect of fiber array, clustered vs separated, is most prominent at relatively low fiber volume fractions, around 50 to 60%. At high volume fractions, say about 60 to 70% or higher, any effects of fiber array are much attenuated, as the fibers must be so close together at high volume fractions that all arrays tend to the clustered kind. The wetting or lack of wetting between matrix and fiber affects clustering, with lack of wetting promoting fiber-fiber contacts.

We have made some progress in understanding how fiber-array details affect shrinkage. The work involved the following elements:

1. Microstructural examination of various samples before and after carbonization, with a view to characterizing clustering, contact networks, and microcracking in relation to observed shrinkage, emphasizing image-analysis measures for the descriptive quantities of interest.
2. Direct observations in a hot-stage microscope of the carbonization process within fiber bundles.
3. Some exploration of three-dimensional aspects of fiber arrays

The results may prove useful to those who might pursue further aspects of this subject, including perhaps questions like:

> How is fiber packing affected by fiber section shape and size variations? by deviations from fiber parallelism? How can packing be quantified and its effects on shrinkage be modeled? Is carbon-volume yield of matrix affected by packing and fiber content? How important are fiber-fiber contact deformations and stresses? Do contact stresses and deformations influence graphitizability of inter-fiber matrix? How do filler additions affect matrix and composite shrinkage?

Of particular interest to the selection of constituents and processes is the question: what are the factors that determine whether and how (and in what relative degrees) matrix shrinkage leads to composite shrinkage, composite porosity, microdamage, or composite delamination?

APPENDIX 1

ACADEMIC MATTERS AND PUBLICATIONS

STUDENTS and THESES

A student at Clarkson University, Mr. Tad Guski, participated in the research under the guidance of Professor Yurgartis.

The following MS thesis was partially supported by this research: Micromechanical Modeling of Transverse Tensile Failure in 2-D Carbon-Carbon Composites, Joel Hughes, Clarkson University, June 22, 1994

PAPERS, PRESENTATIONS, AND PUBLICATIONS

The following publications, presentations, and papers derive, at least in part, from this project:

S. W. Yurgartis, K. Morey, and J. Jortner, Measurement of Yarn Shape and Nesting in Plain-Weave Composites, Comp. Sci. Techn., Vol 46, pp 39-50, 1993.

J Jortner and S W Yurgartis, Dimensional Changes During Manufacture of C/C Composites, presentation to AFOSR-URI Symposium on C/C Composites, Penn State University, Sept 1994.

J Jortner, On Property Scatter of Carbon-Carbon Laminates: Causes, Effects, and Potential for Reduction" invited keynote lecture to American Carbon Society Workshop on Carbon Materials for Advanced Technologies, Oak Ridge, Tennessee, May 1994.

S W Yurgartis, Techniques for the Quantification of Composite Mesostructure, Composites Science and Technology, vol. 53, pp. 145--154 (1995).

J Jortner and S W Yurgartis, Relation of Fiber Contact Arrays to Carbonization Shrinkage, Extended Abstracts of the 22nd Biennial Carbon Conference, American Carbon Society, 1995, pp76-77.

J Jortner, Scatter of Engineering Properties of Carbon-Carbon Composites, presented to the 2nd French-US Meeting for Technical Interchange on C/C Composites and Related High Temperature Materials, La Jolla, CA, March 1995.

J Jortner, Effective Stiffness of Microcracked Components in 3D Carbon-Carbon Composites," Ext. Abstr. 22nd Bienn. Conf. Carbon, American Carbon Society, 1995, pp114-115.

APPENDIX 2

SOFTWARE

Source files are presented for selected image-analysis software developed under this research study. Most of the source files are annotated sufficiently to inform about their purpose and approach. Sources and executable files (which should operate on IBM-style personal computers under DOS) of most of these programs may be obtained on disc (CD or diskette) from Julius Jortner at PO Box 219, Cloverdale, OR 97112, phone: 503-392-4001 or mailto://jjortner@oregoncoast.com.

Program BEN.BAS

Ben Yurgartis, August 1994
prelim description added by J Jortner

```
'Summary by Jortner: This program populates an area with a specified number
'of circles placed at random, without overlap, and with the option of
'providing a specified space between nearest circles.  Once the specified
'number of fibers has been placed, the program allows them to settle
'downwards until contact stops each from moving further.  At that time,
'an option allows the circles to be jogged to the right or to the left and
'further settling may occur.  The program stops when further such movements
'of circles cannot occur.  The graphic output shows clearly what is going on.
'The program is quite clearly annotated - further detail is provided below.


DECLARE SUB Jogg (NumFib!, Center!(), r!, q!, Rad!)
CLS
SCREEN 12
WINDOW

PRINT "Program to model fiber settling.  Ben Yurgartis, 11/29/93"
PRINT
RANDOMIZE TIMER
DIM Center(5000, 2) 'Array to record the center position of all fibers

INPUT "Radius of the fibers: ", Rad         'Allows you to enter the radius,
                                            'extra space, and the number
                                            'of fibers wanted.
INPUT "Extra space around the fibers to start: ", Ext
INPUT "Max # Fibers: ", NumFib

CLS
l = 1
DO WHILE l <= NumFib   'Loop that randomly
x = INT(RND * 640)     'places the number
y = INT(RND * 480)     'of fibers specified.

FOR a = 1 TO l
temp = (x - Center(a, 1)) ^ 2 + (y - Center(a, 2)) ^ 2  'This Loop
b = INT(SQR(temp))                                         'checks to see
IF b < ((Rad + Ext) * 2) THEN GOTO Skip                 'if the current fiber
NEXT a                                                  'interferes with another.

IF (y + (Rad + Ext)) > 480 THEN GOTO Skip   'Checks to make sure
IF (x + Rad) > 640 THEN GOTO Skip           'the fiber is not
IF x < 0 + Rad THEN GOTO Skip               'outside of the screen boundaries

    Center(l, 1) = x
    Center(l, 2) = y
    CIRCLE (x, y), Rad, 4       'Draws and paints the fibers
    PAINT (x, y), 4
    l = l + 1              'This is the number of fibers currently drawn.
Skip:
LOOP

Flag = 1
DO WHILE Flag = 1          'Loop that settles fibers. The Flag stops being
Flag = 0                   ' one when no fiber gets moved down.
FOR r = 1 TO NumFib        'The number of the fibers being moved. Trys to
Jog = 0                    'move all the fibers in each settling loop.
x = Center(r, 1)
y = (Center(r, 2) + 5)     'Changes the y position +5 (downward).
FOR q = 1 TO NumFib        'Loop to check if the movement of the r-fiber
IF q = r THEN q = q + 1    'will interfer with any of the others.

IF Rad * 2 > INT(SQR(((x - Center(q, 1)) ^ 2 + (y - Center(q, 2)) ^ 2))) THEN
        Jog = 1
        GOTO Nextr
END IF
NEXT q                 'The above line checks to see that the fiber does not
```

```
                        'overlap with any other fibers.
IF (Center(r, 2) + 5) + Rad > 480 THEN GOTO Nextr   'Keeps the fibers from
                                                    'going off the screen.
    CIRCLE (Center(r, 1), Center(r, 2)), Rad, 0         'Paints over the old
    PAINT (Center(r, 1), Center(r, 2)), 0               'circle.
    CIRCLE (Center(r, 1), (Center(r, 2) + 5)), Rad, 3      'Draws new circle.
    Center(r, 2) = (Center(r, 2) + 5)       'Changes old y position to new
    Flag = 1                                'y position.

Nextr:
IF Jog = 1 THEN CALL Jogg(NumFib, Center(), r, q, Rad)   'If the fiber will
NEXT r                                                   'not move in the y
LOOP                                                     'direction, it tries
                                                         'to move it in the x.
FOR w = 1 TO NumFib
CIRCLE (Center(w, 1), Center(w, 2)), Rad, 3      'Redraws circles, and then
PAINT (Center(w, 1), Center(w, 2)), 3            'fills them in.
NEXT w
LOCATE 1, 1: INPUT "Press enter to continue...", a$
END

'When a fiber can not move down because it contacts a fiber below, this
'subroutine will move the fiber to the left or right a small amount.
'
SUB Jogg (NumFib, Center(), r, q, Rad)

IF Center(r, 1) - Center(q, 1) < 0 THEN b = -2 ELSE b = 2  'Decide whether to
y = Center(r, 2)                                           'move to the right
x = (Center(r, 1) + b)                                     'or left.
IF x > (640 - Rad) THEN EXIT SUB                   'Keeps fibers wihin the
IF x < (0 + Rad) THEN EXIT SUB                     'screen boundaries.
FOR h = 1 TO NumFib
IF h = r THEN GOTO Jim
temp = ((x - Center(h, 1)) ^ 2 + (y - Center(h, 2)) ^ 2)
t = INT(SQR(temp))
IF t < ((Rad) * 2) THEN EXIT SUB
Jim:
NEXT h
CIRCLE (Center(r, 1), Center(r, 2)), Rad, 0        'Redraws fiber in its
PAINT (Center(r, 1), Center(r, 2)), 0              'new position.
CIRCLE ((Center(r, 1) + b), Center(r, 2)), Rad, 3
Center(r, 1) = (Center(r, 1) + b)
END SUB
```

Code for contact angle and contact density measurements

S W Yurgartis et al, January 1994

The code for contact-angle and contact-density measurements on images of fiber cross-section arrays was written in "C" at Clarkson University. The source file may be obtained from Prof S W Yurgartis, MIE Dept. A brief description follows:

This program will loop around for every on-screen fiber cross-section and then trace around the contour of that fiber. While it traces, the program will search for other fibers in close proximity and record the grey level value of the contacting fiber and the coordinates of the contact point.

Individual contact points between any two specific fibers are used to calculate a centroid which is considered to be the principal fiber/fiber contact point. Contact angles and the number of principal fiber/fiber contact points are extracted from the data arrays.

Program PROXCIRC Version 1 Julius Jortner, Sep 95

```
'Populates a rectangular area with reddish colored circles that do not
'overlap.  To produce a "natural" arrangement within the rectangle,
'circles are placed randomly outside as well as inside the rectangle
'but the ones outside are not counted as to any statistics.

'Contacts are graphically indicated by center-to-center
'lines connecting the centers of contacting circles.

'From an earlier version, there is coding that allows overlap of
'circles to the extent indicated by the FDERR input.  In this current
'code version, FDERR is taken as zero.  However, some minor overlap
'(up to one pixel) is allowed via an adjustment to RFO, the radius of
'the exclusion circle; otherwise, contact never occurs.

'This version displays on a standard VGA screen.

'It is a PRELIMINARY version.
'Please report bugs to jjortner@oregoncoast.com or (503)392-4001.
'Using subroutines would make the code easier to read, but I have
'had some problems getting subs to work in this version of BASIC.

'For illustrative purposes, if requested, the code will plot
'a regular hexagonal array, an open hexagonal array, or a regular
'square array, at an area fraction input by the user.

'The circle-placement image displays fairly quickly.  The various
'calculations take some time before the results appear.

'The area fraction of circles is calculated from the colored pixels.
'(A cruder area estimate based on counting circle cross-sections also
'is provided.)

'The attained area fraction of randomly placed circles depends, in a
'random sort of way, on the number of iterations requested before the
'program gives up on finding a space for the new circle.  The left-over
'space available for circle center-point placement is easily seen on
'the final display; it is the uncolored (black) areas within the
'counting-space rectangle.
'
'The proximity distance (the perimeter-to-perimeter distance) is
'calculated for each of the six circles closest to each of the
'full-area circles.  After all the proximity distances are sorted
'in ascending order, the proximity-distance statistics are plotted
'as a cumulative probability ogive; the ordinate is the estimated
'probability of finding a proximity distance less than indicated on
'the abscissa.  One unit on the abscissa scale is the proximity distance
'of nearest neighbors in a regular hexagonal array of the same area

'Observations about typical results include:

'(1) The calculations for the regular hexagonal array produce data on
'a vertical line at the expected position.  The calculations for the
'open hexagonal array and the square array also produce the expected
'plots.  This tends to validate the proximity calculations.
'(2) The area fraction computed from pixel colors is significantly
'larger than input for the regular array.  This may be problem with
'whole-pixel representation of circles and the fact that the counting
'area is not necessarily an integer multiple of the area per circle?
'(3) The plot for random placement tends to be nearly linear, proceeding
'from the origin approximately through the hexagonal-array vertical at
'the 50% point.


DEFINT I-N:
OPTION BASE 1
CLS
SCREEN 12
```

```
PRINT "CIRCULAR FIBER PLACEMENT MODEL, PROXIMITY VERSION 1.0"
PRINT
'trial inputs for checkout
'==================================================
NM = 500: XMX = 400: YMX = 400: DB = 60: DF = 30: FDERR = 0
'instead of inputs supplied at runtime
'INPUT "DIMENSION OF COUNTING-SPACE BORDER: ", DB
'INPUT "AVERAGE FIBER DIAMETER: ", DF
'==================================================
PRINT "RANDOM ARRAY?           IF 'YES'  ENTER 0: "
PRINT "REGULAR HEX ARRAY?      IF 'YES'  ENTER 1: "
PRINT "REGULAR OPEN HEX ARRAY? IF 'YES'  ENTER 2: "
PRINT "REGULAR SQUARE ARRAY?   IF 'YES'  ENTER 3: "
INPUT "Enter Choice Here.......................: ", KHEX
IF KHEX = 0 THEN
        'INPUT "TOLERANCE DEFINING CONTACT (decimal fraction): ", FDERR
        INPUT "MAX NUM OF ITERATIONS RE INTERFERENCE: ", NITERMX
ELSE
        INPUT "DESIRED AREA FRACTION IS : ", AFD
END IF

DIM XF(400), YF(400), ICOUNT(400), NC(400), JCONT(7), HCONT(7)
DIM PROX(400, 36), CDIST(400, 36), DISTRY(400), NPROX(400), TPROX(2000)

PI = 3.14159
DERR = FDERR * DF
RF = DF / 2:
RFO = 2 * RF - DERR - 1             'radius of exclusion zone is
                                                  'reduced by one pixel to allow
                                                  'some random contact
AFIBER = PI * RF ^ 2
PFACT = 2

X1 = DB                   'define frame for area and contact counting
X2 = XMX - DB
Y1 = DB
Y2 = YMX - DB
ASPACE = (X2 - X1) * (Y2 - Y1)

XIN1 = X1 + RF - DERR      'define frame for fully countable fibers
XIN2 = X2 - RF + DERR      '  (entire circle within counting space)
YIN1 = Y1 + RF - DERR
YIN2 = Y2 - RF + DERR

XO1 = X1 - RF              'define frame for partly countable fibers
XO2 = X2 + RF              '   (part circle within counting space)
YO1 = Y1 - RF
YO2 = Y2 + RF

CLS
JCOLBOX = 7                           'color for counting_area box
JCOLOVER = 3                          'color for overlap circles
JCOLCIRC = 5                          'color for placed circles
JCOLCONT = 8                          'color for contact links
JCOLOVER1 = 9                         'color: overlap circle border
JCOLCIRC1 = 6                         'color: placed circle border

'------------------------------ place fibers and check for overlap
IF KHEX = 0 THEN
NCONT = 0: NFULL = 0: NPART = 0
RANDOMIZE TIMER
FOR I = 1 TO NM
NITER = 0
Newil:
        X = RND * XMX:                          'locate center of fiber i
        Y = RND * YMX:
        IF POINT(X, Y) <> 0 THEN
                 NITER = NITER + 1
                 IF NITER > NITERMX THEN GOTO results '- escape if too many tries
                 GOTO Newil                          '- try again if interference
        END IF
```

```
        XF(I) = X
        YF(I) = Y
        CIRCLE (X, Y), RFO, JCOLOVER1
        PAINT (X, Y), 0, JCOLOVER1
        PAINT (X, Y), JCOLOVER, JCOLOVER1
        CIRCLE (X, Y), RFO, JCOLOVER
'     -----------------------------------------------------------
'     find whether full area of fiber is in counting space
'     ICOUNT(I)=1 if it's fully within the counting frame
'     ICOUNT(I)=2 if it's only partly within counting frame
'     ICOUNT(I)=0 if it isn't at all within counting frame
'     (Counting frame defined by (X1,Y1)-(X2,Y2))
        IF X >= XIN1 AND X <= XIN2 AND Y >= YIN1 AND Y <= YIN2 THEN
                ICOUNT(I) = 1: NFULL = NFULL + 1
        ELSEIF X > XO1 AND X < XO2 AND Y > YO1 AND Y < YO2 THEN
                ICOUNT(I) = 2: NPART = NPART + 1
        ELSE
                ICOUNT(I) = 0
        END IF
'     -----------------------------------------------------------
        NF = I
NEXT I
'------------------------------------------------------------
END IF

'------------------------------------------ calc stats for hexagonal array
IF KHEX <> 0 THEN                           ' at input area fraction
        AHEXAGON = 3 * AFIBER / AFD:
        ANGLE = 30 * PI / 180: COSFACT = COS(ANGLE)
        DC = SQR(AHEXAGON / (3 * COSFACT))
        HEXPROX = (DC - DF)
END IF
'--------------------------------------------------------------------------
'------------------------------------------ draw hexagonal array of circles
IF KHEX = 1 THEN
IF DC < DF THEN PRINT "Requested Area Fraction Too High!": END
HXSTEPX = DC
HXSTEPY = COSFACT * HXSTEPX
IROW = -1
I = 1
Y = 1
DO WHILE Y < YMX
        IROW = -1 * IROW
        IF IROW < 0 THEN
                X = HXSTEPX / 2 + 1:
        ELSE
                X = 1
        END IF
        DO WHILE X < XMX
                XF(I) = X: YF(I) = Y
                IF X >= XIN1 AND X <= XIN2 AND Y >= YIN1 AND Y <= YIN2 THEN
                        ICOUNT(I) = 1: NFULL = NFULL + 1
                ELSEIF X > XO1 AND X < XO2 AND Y > YO1 AND Y < YO2 THEN
                        ICOUNT(I) = 2: NPART = NPART + 1
                ELSE
                        ICOUNT(I) = 0
                END IF
                I = I + 1
                X = X + HXSTEPX
        LOOP
        Y = HXSTEPY + Y
LOOP
NF = I
END IF
'------------------------------------------------------------
'------------------------------------------ draw open hexagonal array
IF KHEX = 2 THEN
PFACT = 3.5
AHEXAGON = 2 * AFIBER / AFD:
ANGLE = 30 * PI / 180: COSFACT = COS(ANGLE)
DC = SQR(AHEXAGON / (3 * COSFACT))
```

```
IF DC < DF THEN PRINT "Requested Area Fraction Too High!": END
TRIPROX = (DC - DF)
B = 2 * DC * COSFACT
TSTEPX = B
TSTEPY = COSFACT * TSTEPX
IROW = 1:
I = 1
Y = 1
DO WHILE Y < YMX
        IF IROW = 1 THEN
                X = 1
                TSTEPY = DC / 2
        ELSEIF IROW = 2 THEN
                X = 1 + B / 2
                TSTEPY = DC
        ELSEIF IROW = 3 THEN
                X = 1 + B / 2
                TSTEPY = DC / 2
        ELSEIF IROW = 4 THEN
                X = 1
                TSTEPY = DC
        END IF
        IROW = IROW + 1
        IF IROW = 5 THEN IROW = 1
        DO WHILE X < XMX
                XF(I) = X: YF(I) = Y
                IF X >= XIN1 AND X <= XIN2 AND Y >= YIN1 AND Y <= YIN2 THEN
                        ICOUNT(I) = 1: NFULL = NFULL + 1
                ELSEIF X > XO1 AND X < XO2 AND Y > YO1 AND Y < YO2 THEN
                        ICOUNT(I) = 2: NPART = NPART + 1
                ELSE
                        ICOUNT(I) = 0
                END IF
                I = I + 1
                X = X + TSTEPX
        LOOP
        Y = TSTEPY + Y
LOOP
NF = I
END IF
'------------------------------------------------------------
'------------------------------------------draw square array of circles
IF KHEX = 3 THEN
PFACT = 1.5
ASQUARE = AFIBER / AFD
DC = SQR(ASQUARE)
IF DC < DF THEN PRINT "Requested Area Fraction Too High!": END
SQPROX = DC - DF
X = 1
Y = 1
I = 1
DO WHILE Y < YMX
        DO WHILE X < XMX
                XF(I) = X: YF(I) = Y
                IF X >= XIN1 AND X <= XIN2 AND Y >= YIN1 AND Y <= YIN2 THEN
                        ICOUNT(I) = 1: NFULL = NFULL + 1
                ELSEIF X > XO1 AND X < XO2 AND Y > YO1 AND Y < YO2 THEN
                        ICOUNT(I) = 2: NPART = NPART + 1
                ELSE
                        ICOUNT(I) = 0
                END IF
                I = I + 1
                X = X + DC
        LOOP
        X = 1
        Y = Y + DC
LOOP
NF = I
END IF
'------------------------------------------------------------
```

```
results:
'----------------------------------------------------------
FOR I = 1 TO NF                        'draw fiber circles
       LX = (XF(I)): LY = (YF(I))
       CIRCLE (LX, LY), RF, JCOLCIRC1:
       PAINT (LX, LY), 0, JCOLCIRC1
       PAINT (LX, LY), JCOLCIRC, JCOLCIRC1
       CIRCLE (LX, LY), RF, JCOLCIRC:
NEXT I


LINE (X1 - 1, Y1 - 1)-(X2 + 1, Y2 + 1), JCOLBOX, B   'draw counting box

                                                        'For programming purposes only
'INPUT "GRAPH CHECK?:", KGRAPH         'Modify output graph.......
'IF KGRAPH <> 0 THEN GOTO Checkgraph

'---------------------------------------- estimate fiber area fraction
KX1 = X1: KX2 = X2: KY1 = Y1: KY2 = Y2
AREAF = 0
AREAT = (X2 - X1 + 1) * (Y2 - Y1 + 1)
FOR KX = KX1 TO KX2
          FOR KY = KY1 TO KY2
               IF POINT(KX, KY) = JCOLCIRC THEN
                          AREAF = AREAF + 1
               END IF
          NEXT KY
NEXT KX
AF = AREAF / AREAT
AFTRY = (NFULL + NPART / 2) * AFIBER / ASPACE

IF KHEX = 0 THEN                       'regular hex array stats for
       AFIBER = PI * RF ^ 2              'pixel-based area fraction
       AHEXAGON = 3 * AFIBER / AF
       ANGLE = 30 * PI / 180: COSFACT = COS(ANGLE)
       DC = SQR(AHEXAGON / (3 * COSFACT))
       HEXPROX = (DC - DF)
END IF
'----------------------------------------------------------------------

'---------------------------------------- calculate center-center distances
FOR I = 1 TO NF
       NC(I) = 0
       IF ICOUNT(I) = 1 THEN            ' for full-area circles only
       NPROX(I) = 0
       FOR J = 1 TO NF
               IF J <> I THEN
               DSQR = (XF(I) - XF(J)) ^ 2 + (YF(I) - YF(J)) ^ 2
               IF DSQR <= DF ^ 2 THEN LINE (XF(I), YF(I))-(XF(J), YF(J)), JCOLCONT
               '---------------------------- proximity calculation
               DISTRY = SQR(DSQR)
               IF DISTRY <= PFACT * DC THEN
                      NPROX(I) = NPROX(I) + 1
                      CDIST(I, NPROX(I)) = DISTRY
                      PROX(I, NPROX(I)) = DISTRY - DF
                      IF PROX(I, NPROX(I)) < 0 THEN PROX(I, NPROX(I)) = 0
               END IF
               END IF
       NEXT J
       END IF
NEXT I

'-------------------------------------calculate proximity statistics
FOR I = 1 TO NF
       IF ICOUNT(I) = 1 THEN     'consider only full-area fibers
       FLIPS = -1
       WHILE FLIPS               'sort PROX in ascending order
               FLIPS = 0
               FOR KP = 2 TO NPROX(I)
                IF PROX(I, KP - 1) > PROX(I, KP) THEN
                      FLIPS = -1
```

```
                        SWAP PROX(I, KP - 1), PROX(I, KP)
                 END IF
                NEXT KP
        WEND
        END IF
NEXT I
J = 0
FOR I = 1 TO NF
        IF ICOUNT(I) = 1 THEN
        FOR KP = 1 TO 6                      'consider only closest 6 proximities
                J = J + 1
                TPROX(J) = PROX(I, KP) / HEXPROX   'and normalize to HEXPROX
        NEXT KP
        END IF
NEXT I

'-----------------------------------sort TPROX in ascending order
FLIPS = -1
WHILE FLIPS
         FLIPS = 0
         FOR KP = 2 TO J
                 IF TPROX(KP - 1) > TPROX(KP) THEN
                         FLIPS = -1
                         SWAP TPROX(KP - 1), TPROX(KP)
                 END IF
         NEXT KP
WEND
'---------------------------------plot proximity ogive
LINE (XMX - 1.1 * DB / 2, 1)-(639, 479), 1, B
PAINT (XMX + DB, 200), 3, 1
PAINT (XMX + DB, 200), 0, 1
Checkgraph:
        LOCATE 1, 47: PRINT "Perimeter-Perimeter Distance"
        LOCATE 2, 47: PRINT "  to 6 Nearest Neighbors"
        LOCATE 3, 47: PRINT "   CUMULATIVE FREQUENCY"
        LOCATE 4, 47: PRINT "                         "
        LOCATE 13, 47: PRINT "0.5"
        LOCATE 22, 47: PRINT "  0"
        LOCATE 23, 49: PRINT "  0"
        LOCATE 4, 47: PRINT "1.0"
        LOCATE 23, 58: PRINT "  1"
        LOCATE 23, 67: PRINT "   2"
        LOCATE 23, 76: PRINT "   3"
        LOCATE 24, 51: PRINT " PROXIMITY DISTANCE RELATIVE"
        LOCATE 25, 51: PRINT "TO NEAREST 6 IN REG HEX ARRAY"
        LINE (XMX, DB)-(630, YMX - DB), JCOLOVER1, B
        PAINT (XMX + 5, DB + 5), 6, JCOLOVER1
        YORD = YMX - 2 * DB: XORD = 630 - XMX
        KXSCALE = 3: KYSCALE = J
        FOR ILINE = 0 TO KXSCALE
        KXLINE1 = XMX + ILINE * XORD / KXSCALE
        LINE (KXLINE1, YMX - DB)-(KXLINE1, DB), JCOLOVER1
        NEXT ILINE
        KYLINE1 = YMX / 2
        LINE (XMX, KYLINE1)-(630, KYLINE1), JCOLOVER1
        FOR KP = 1 TO J
        KY = (KP / KYSCALE) * YORD: KX = (TPROX(KP) / KXSCALE) * XORD
        IF KP = 1 THEN
                PSET (KX + XMX, YMX - DB - KY), JCOLOVER
        ELSE
                LINE STEP(0, 0)-(KX + XMX, YMX - DB - KY), JCOLOVER
        END IF
        NEXT KP

        LOCATE 1, 1                   'Checkout feature***************
        PRINT "No. data @ 6 per full fiber: "; : PRINT USING "####"; J

LOCATE 27, 1
SP$ = "      "
IF KHEX <> 0 THEN
        PRINT "REQUESTED AREA FRACTION    : ";
```

```
        PRINT USING "#.###"; AFD;
        PRINT SP$;
ELSE
        PRINT "MAX No PLACEMENT ITERATIONS: ";
        PRINT USING "#####"; NITERMX;
        PRINT SP$;
END IF
PRINT "NUMBER OF FIBERS PLACED     : ";
PRINT USING "#####"; NF;
PRINT SP$;
PRINT "NUMBER OF FULL-AREA FIBERS : ";
PRINT USING "#####"; NFULL;
PRINT SP$;
PRINT "NUMBER OF PART-AREA FIBERS : ";
PRINT USING "#####"; NPART;
PRINT SP$;
PRINT "FIBER AREA FRACTION (pixel): ";
PRINT USING "#.###"; AF; : PRINT SP$;
PRINT "FIBER AREA FRACTION (crude): ";
PRINT USING "#.###"; AFTRY; : PRINT SP$;
'-----------------------------------------------------------
EndLoop:
IF INSTAT THEN END
GOTO Endloop
'-----------------------------------------------------END
```

```
'Program CIRCSEG    Version 2                  Julius Jortner, Oct 95

'GIVES LINE-SEGMENT STATISTICS FOR RANDOM AND REGULAR CIRCLE ARRAYS
'PROVIDES FOR READING REAL MICROGRAPH.................BIF FORMAT
'      see notes at end of program (after subroutines)
DEFINT I-N:
OPTION BASE 1
CLS
SCREEN 12

PRINT "CIRCULAR FIBER PLACEMENT MODEL, Line-Segment VERSION 1.0"
PRINT

'====================================================
NM = 400: XMX = 440: YMX = 440: DB = 88: DF = 40: FDERR = 0
PRINT "REAL MICRO FROM FILE?   IF 'YES' ENTER -1: "
PRINT "RANDOM ARRAY?           IF 'YES'  ENTER 0: "
PRINT "REGULAR HEX ARRAY?      IF 'YES'  ENTER 1: "
PRINT "REGULAR OPEN HEX ARRAY? IF 'YES'  ENTER 2: "
PRINT "REGULAR SQUARE ARRAY?   IF 'YES'  ENTER 3: "
INPUT "Enter Choice Here........................: ", KARRAYTYPE
IF KARRAYTYPE = 0 THEN
        INPUT "MAX NUM OF ITERATIONS RE INTERFERENCE: ", NITERMX
        INPUT "RELATIVE CONTACT TOLERANCE           : ", FDERR
ELSEIF KARRAYTYPE > 0 THEN
        INPUT "DESIRED AREA FRACTION IS : ", AREAINP
        INPUT "RANDOM ORIENTATION? IF SO, ENTER 1: ", KORIENT
END IF
INPUT "NEED AREA FRACTION STATS? IF 'YES' ENTER 1: ", KAREA
PRINT
PRINT "How do you want to normalize segment lengths?"
PRINT "TO REG. HEX. PROXIMITY?  IF YES, ENTER 1"
PRINT "TO REG. HEX. CENTER-CENTER DIST? ENTER 0"
INPUT "Enter choice here......................: ",KNORMAL

DIM XF(400), YF(400), ICOUNT(400), NC(400), JCONT(7), HCONT(7)
'DIM PROX(400, 36), CDIST(400, 36), DISTRY(400), NPROX(400), TPROX(4000)
DIM LCIRC(6000), LMATR(6000), OTITLE$(24)
```

```
PI = 3.14159
RF = DF / 2
DERR=FDERR*DF
RFO = 2 * RF-DERR - 1                'radius of exclusion zone is
'                                     reduced by one pixel to allow
'                                     some random contact
AFIBER = PI * RF ^ 2
NCONT = 0: NFULL = 0: NPART = 0
'-----------------------------------------------------------------------
JCOLBOX =   7                              'color for counting_area box
JCOLOVER =  3                             'color for overlap circles
JCOLCIRC =  5                             'color for placed circles
JCOLCONT =  8                             'color for contact links
JCOLOVER1 = 9                            'color: overlap circle border
JCOLCIRC1 = 4                            'color: placed circle border

CLS
'-------------------------------------------read real image file & draw
IF KARRAYTYPE = -1 THEN
        CALL REALMICRO(DF,AREAFRAC)
        RF=DF/2
        AREAINP=AREAFRAC
    X1=10:X2=500:Y1=10:Y2=470
    LINE (X1-3,Y1-3)-(X2+3,Y2+3),JCOLBOX,B
END IF
'------------------------------------------ calc stats for hexagonal array
IF KARRAYTYPE <> 0 THEN
   CALL HEXSTATS(RF,AREAINP,HEXPROX,DC)
'   print hexprox,dc,rf:stop
END IF
'------------------------------------------define circle center locations
IF KARRAYTYPE=0 THEN
   PFACT=2                                                  'random array
   CALL RNDARRAY (NM,RFO,XMX,YMX,NITERMX,JCOLOVER1,JCOLOVER,NF,XF(),YF())
ELSEIF KARRAYTYPE=1 THEN
   PFACT=1.2:RORANGE=60
   CALL HEXARRAY (DC,DF,XMX,YMX,XF(),YF(),NF)               'hexagonal
ELSEIF KARRAYTYPE=2 THEN
   PFACT=3.5:RORANGE=60
   CALL TRIARRAY (AREAINP,DC,DF,AFIBER,XMX,YMX,XF(),YF(),NF)  'open hexagonal
ELSEIF KARRAYTYPE=3 THEN
   PFACT=1.5:RORANGE=90
   CALL SQARRAY (AREAINP,DC,DF,AFIBER,XMX,YMX,XF(),YF(),NF)   'square
END IF
IF KORIENT<>0 AND KARRAYTYPE<>0 THEN             'rotate regular array
        XC=XMX/2                                 'thru random angle
        YC=YMX/2                                 'for sake of random
        CALL ROTAXY(NF,XC,YC,RORANGE,XF(),YF())  'scanning of the image
END IF                                           'with horiz/vert scans'-----------------
--------------------------------------------------------------
IF KARRAYTYPE<>-1 THEN
CALL CLASS(XMX,YMX,DB,RF,DERR,NF,XF(),YF(),ASPACE,ICOUNT(),X1,X2,Y1,Y2,NFULL,NPART)
END IF
'-------------------------------------------'classify center loci
IF KARRAYTYPE >=0 THEN
        CLS
        CALL DRAWCIRC (NF,XF(),YF(),RF,JCOLCIRC,JCOLCIRC1)      'draw circles
        LINE (X1 - 1, Y1 - 1)-(X2 + 1, Y2 + 1), JCOLBOX, B   'draw counting box
END IF

CALL PIXAREA (X1,X2,Y1,Y2,JCOLCIRC,AREAPIX)          'area fraction via pixels
CALL POINTAREA (X1,X2,Y1,Y2,JCOLCIRC,AREAPOINTS)     'area fraction via points
'CALL COUNTAREA (NFULL,NPART,AFIBER,ASPACE,AFTRY)     'area fraction via count
                                                                         'area
fraction via lines

CALL LINEMATR(X1,X2,Y1,Y2,JCOLCIRC,JCOLCIRC1,AREALINES,LMATR(),NSEG2)

IF KARRAYTYPE = 0 THEN
        CALL HEXSTATS(RF,AREAPIX,HEXPROX,DC)
```

```
ELSE
        RFADJ=RF*SQR(AREAPIX/AREAINP)             'adjust circle radius for
    CALL HEXSTATS(RFADJ,AREAPIX,HEXPROX,DC)     'pixel circle errors
END IF

LOCATE 19,1         '-------------------------print analysis statistics
print "LINE-SEGMENT ANALYSIS AND AREA FRACTIONS"
IF KARRAYTYPE=0 THEN
        print "# placement iterations: ";:print using "######";nitermx
ELSE
        print "input area fraction:  "; :print using "#.####";areainp
END IF
'print "# segmts in circles: ";:print using "######";nseg1
print "# segmts in matrix : ";:print using "######";nseg2
print "radius adjustment:   ";:print using "#.####";rfadj/rf
print "linesegmts areafrac: ";:print using "#.####";arealines
IF KAREA=1 THEN
        print "pixel area fraction: ";:print using "#.####";areapix
        print "point area fraction: ";:print using "#.####";areapoints
        print "circle count areafr: ";:print using "#.####";aftry
END IF
print "# full-area circles: ";:print using "######";nfull
print "# part-area circles: ";:print using "######";npart

ARRAY SORT LMATR() FOR NSEG2          'sort LMATR in ascending order

                                      'set up to print graph
        OTITLE$(1)= "Line-Segment Lengths for Matrix"
        OTITLE$(2)= "                               "
        OTITLE$(3)= "    CUMULATIVE FREQUENCY"
        OTITLE$(4)= "                    "
        OTITLE$(5)= " SEGMENT LENGTH RELATIVE"
IF KNORMAL=1 THEN
        OTITLE$(6)= "    TO MIN HEX PROXIMITY "
        CALL OGIVE(XSCALE,YSCALE,NSEG2,LMATR(),HEXPROX,OTITLE$())
ELSE
        OTITLE$(6)= " TO HEX CENTER DISTANCE "
        CALL OGIVE(XSCALE,YSCALE,NSEG2,LMATR(),DC,OTITLE$())
END IF

EndLoop:
IF INSTAT THEN END
GOTO Endloop
'----------------------------------------------------END

'FUNCTIONS AND PROCEDURES
'=====================================================================
DEF FNHYPO(X,Y)=SQR(X^2+Y^2)      'calculates hypotenuse via Pythagoras
'=====================================================================
DEF FNARCTAN(X,Y)                 'returns the angle in radians between
    pi=3.14159                    'the vector from the origin to point (x,y)
    zerotol=.00001                'and the x-axis
    if x<zerotol and x>-zerotol then
          if y>0 then
               alpha=pi/2
          else
               alpha=3*pi/2
          end if
    else
       alpha=atn(y/x)
    end if
    if x<0 then
       alpha=alpha+pi
       elseif x>=0 and y<0 then
       alpha=alpha+2*pi
    end if
    FNARCTAN=ALPHA
END DEF
'=====================================================================
SUB ROTAXY(N,XC,YC,RORANGE,X(),Y())
```

```
'Finds new values for N coordinate pairs X,Y via rotation around
'point XC,XC through an angle BETAD which is selected
'randomly in the range between 0 and RORANGE degrees

pi=3.14159
RANDOMIZE TIMER
BETAD=RND*RORANGE
BETAR=PI*BETAD/180
FOR I=1 TO N
    XX=X(I)-XC
    YY=Y(I)-YC
        R=fnHYPO(XX,YY)
    ALPHAR=fnARCTAN(XX,YY)
    GAMMAR=BETAR+ALPHAR
    X(I)=R*COS(GAMMAR)+XC
    Y(I)=R*SIN(GAMMAR)+YC
NEXT I

END SUB
'=======================================================================
SUB CLASS(XMX,YMX,DB,RF,DERR,NF,XF(),YF(),ASPACE,ICOUNT(),X1,X2,Y1,Y2,NFULL,NPART)

'classifies circles according to center location on screen

X1 = DB                    'define frame for area and contact counting
X2 = XMX - DB
Y1 = DB
Y2 = YMX - DB
ASPACE = (X2 - X1) * (Y2 - Y1)

XIN1 = X1 + RF - DERR       'define frame for fully countable fibers
XIN2 = X2 - RF + DERR       '  (entire circle within counting space)
YIN1 = Y1 + RF - DERR
YIN2 = Y2 - RF + DERR

XO1 = X1 - RF               'define frame for partly countable fibers
XO2 = X2 + RF               '   (part circle within counting space)
YO1 = Y1 - RF
YO2 = Y2 + RF

FOR I=1 TO NF  '    -------------------------------------------------------
'    find whether full area of fiber is in counting space
'    ICOUNT(I)=1 if it's fully within the counting frame
'    ICOUNT(I)=2 if it's only partly within counting frame
'    ICOUNT(I)=0 if it isn't at all within counting frame
'    (Counting frame defined by (X1,Y1)-(X2,Y2))
        X=XF(I):Y=YF(I)
        IF X >= XIN1 AND X <= XIN2 AND Y >= YIN1 AND Y <= YIN2 THEN
                ICOUNT(I) = 1: NFULL = NFULL + 1
        ELSEIF X > XO1 AND X < XO2 AND Y > YO1 AND Y < YO2 THEN
                ICOUNT(I) = 2: NPART = NPART + 1
        ELSE
                ICOUNT(I) = 0
        END IF
NEXT I
END SUB
'=======================================================================
SUB DRAWCIRC (N,X(),Y(),R,JCOL,JCOLB)
'draws N circles of radius R at centers X,Y
'and paints them in color JCOL
FOR I = 1 TO N
        LX = (X(I))
           LY = (Y(I))
        CIRCLE (LX, LY), R, JCOLB
        PAINT (LX, LY), JCOL, JCOLB
NEXT I
END SUB
'=====================================================================
SUB POINTAREA (X1,X2,Y1,Y2,JCOL1,AREA1)

'Estimates area fraction AREA1 of color JCOL1 within a rectangle
```

```
'defined by (X1,Y1)-(X2,Y2), using the method of random point
'placement.  The number of points checked is NPOINTSMAX

NPOINTSMAX=2000
RANDOMIZE TIMER
DO WHILE NPOINTS <= NPOINTSMAX
        X=RND*(X2-X1)+X1
        Y=RND*(Y2-Y1)+Y1
        JCOL=POINT (X,Y)
        IF JCOL<>0 THEN NPOINT1=NPOINT1+1
        NPOINTS=NPOINTS+1
LOOP
AREA1=NPOINT1/NPOINTS

END SUB
'===========================================================================
SUB PIXAREA (X1,X2,Y1,Y2,JCOLCIRC,AF)
'estimate area fraction AF of a phase colored JCOL
'within a rectangular area defined by (X1,Y1)-(X2,Y2)
'using pixel counting over a full scan of the area

KX1 = X1: KX2 = X2: KY1 = Y1: KY2 = Y2
AREAF = 0
AREAT = (X2 - X1 + 1) * (Y2 - Y1 + 1)
FOR KX = KX1 TO KX2
           FOR KY = KY1 TO KY2
                IF POINT(KX, KY) <> 0 THEN
                           AREAF = AREAF + 1
                END IF
           NEXT KY
NEXT KX
AF = AREAF / AREAT
END SUB
'===========================================================================
SUB COUNTAREA (NFULL,NPART,ACIRC,ASPACE,AF)

'estimates area fraction of circles of area ACIRC
'within a total area ASPACE based on counting
'the NFULL circles that are totally within ASPACE
'and the NPART circles that lie on the border

AF = (NFULL + NPART / 2) * ACIRC / ASPACE

END SUB
'===========================================================================
SUB HEXSTATS (R,AF,HEXPROX,DC)

'calculates the closest perimeter-perimeter distance HEXPROX
'and the center-center distance DC between circles of radius R
'arranged in a regular hexagonal array of area fraction AF

        PI=3.14159
        ACIRC = PI * R ^ 2
           DF=2*R
        AHEXAGON = 3 * ACIRC/AF
        ANGLE = 30 * PI / 180: COSFACT = COS(ANGLE)
        DC = SQR(AHEXAGON / (3 * COSFACT))
           IF DC < DF THEN
                 PRINT "Requested Area Fraction Too High!: ", AF: GOTO EndLoop
           END IF
           HEXPROX = (DC - DF)

END SUB
'==========================================================================
SUB HEXARRAY(DC,DF,XMX,YMX,XF(),YF(),NF)

'draws a hexagonal array of circles of diameter DF
'on centers defined by a center-center distance DC
PI=3.14159
COSFACT = COS(30*PI/180)
HXSTEPX = DC
```

```
HXSTEPY = COSFACT * HXSTEPX
IROW = -1
I = 1
Y = 1
DO WHILE Y < YMX
        IROW = -1 * IROW
        IF IROW < 0 THEN
                X = HXSTEPX / 2 + 1:
        ELSE
                X = 1
        END IF
        DO WHILE X < XMX
                XF(I) = X: YF(I) = Y
'          IF X >= XIN1 AND X <= XIN2 AND Y >= YIN1 AND Y <= YIN2 THEN
'               ICOUNT(I) = 1: NFULL = NFULL + 1
'          ELSEIF X > XO1 AND X < XO2 AND Y > YO1 AND Y < YO2 THEN
'               ICOUNT(I) = 2: NPART = NPART + 1
'          ELSE
'               ICOUNT(I) = 0
'          END IF
                I = I + 1
                X = X + HXSTEPX
        LOOP
        Y = HXSTEPY + Y
LOOP
NF = I
END SUB
'========================================================================
SUB RNDARRAY(NM,RFO,XMX,YMX,NITERMX,JCOL1,JCOL2,NF,XF(),YF())

'locates centers XF,XY of randomly placed nonoverlapping circles
'of diameter DF within a frame defined by XMX,YMX.
'Uses pixel colors to determine overlap.  Attempts NITERMX iterations
'before giving up on placing a circle.

RANDOMIZE TIMER
FOR I = 1 TO NM
NITER = 0
Newil:
        X = RND * XMX:                       'locate center of fiber i
        Y = RND * YMX:
        IF POINT(X, Y) <> 0 THEN
          NITER = NITER + 1
          IF NITER > NITERMX THEN EXIT SUB  '- escape if too many tries
          GOTO Newil                         '- try again if interference
        END IF

        XF(I) = X
        YF(I) = Y
        CIRCLE (X, Y), RFO, JCOL1
        PAINT (X, Y), 0, JCOL1
        PAINT (X, Y), JCOL2, JCOL1
        CIRCLE (X, Y), RFO, JCOL2

        NF = I
NEXT I
'-----------------------------------------------------------
END SUB
'========================================================================
SUB SQARRAY(AREAINP,DC,DF,AFIBER,XMX,YMX,XF(),YF(),NF)
PFACT = 1.5
ASQUARE = AFIBER / AREAINP
DC = SQR(ASQUARE)
IF DC < DF THEN PRINT "Requested Area Fraction Too High!": GOTO EndLoop
SQPROX = DC - DF
X = 1
Y = 1
I = 1
DO WHILE Y < YMX
        DO WHILE X < XMX
                XF(I) = X: YF(I) = Y
```

```
                I = I + 1
                X = X + DC
        LOOP
        X = 1
        Y = Y + DC
LOOP
NF = I
END SUB
'======================================================================
SUB TRIARRAY(AREAINP,DC,DF,AFIBER,XMX,YMX,XF(),YF(),NF)

'for a given area fraction AREAINP, this routine locates the centers XF,XY
'of circles of diameter DF arrayed in an open hexagonal manner
'(each circle has three nearest neighbors) within a rectangular frame
'defined by XMX,YMX.

PI=3.14159
AHEXAGON = 2 * AFIBER / AREAINP
ANGLE = 30 * PI / 180: COSFACT = COS(ANGLE)
DC = SQR(AHEXAGON / (3 * COSFACT))
IF DC < DF THEN PRINT "Requested Area Fraction Too High!": GOTO EndLoop
TRIPROX = (DC - DF)
B = 2 * DC * COSFACT
TSTEPX = B
TSTEPY = COSFACT * TSTEPX
IROW = 1:
I = 1
Y = 1
DO WHILE Y < YMX
        IF IROW = 1 THEN
                X = 1
                TSTEPY = DC / 2
        ELSEIF IROW = 2 THEN
                X = 1 + B / 2
                TSTEPY = DC
        ELSEIF IROW = 3 THEN
                X = 1 + B / 2
                TSTEPY = DC / 2
        ELSEIF IROW = 4 THEN
                X = 1
                TSTEPY = DC
        END IF
        IROW = IROW + 1
        IF IROW = 5 THEN IROW = 1
           DO WHILE X < XMX
                XF(I) = X: YF(I) = Y
                I = I + 1
                X = X + TSTEPX
        LOOP
           Y = TSTEPY + Y
LOOP
NF = I
END SUB
'=======================================================================
SUB LINEMATR(X1,X2,Y1,Y2,JCOL1,JCOL2,AREA1,L2(),I2)

'Scans a rectangular area defined by (X1,Y1)-(X2,Y2)
'with randomly placed vertical and horizontal lines
'to calculate the area fraction AREA1 of pixels of color JCOL1.
'Gathers lengths of line segments totally of background color
'into and array L2. A total of NLINESMX lines are scanned, half
'vertical and half horizontal.


NLINESMX=500
RANDOMIZE TIMER
XRANGE=X2-X1: YRANGE=Y2-Y1
TOTALEN=0 : TOTLEN1=0 : TOTLEN2=0 : I2=0
%X=1 : %Y=0
PH1=JCOL1: BORDER=JCOL2
DO UNTIL NLINES=NLINESMX    '-----------------place random lines
```

```
  IF NLINES>NLINESMX/2 THEN
       X=RND*XRANGE+X1:Y=Y1:IDIR=%X
  ELSE
       Y=RND*YRANGE+Y1:X=X1:IDIR=%Y
  END IF

  XO=X:YO=Y
  JCOLO=POINT(XO,YO)
  ISEG=0 : LSTOP=0
  DO  '****************************** identify segments of color in a line
       DO                              '===============find new-color point
               IF IDIR=%X THEN
                       Y=Y+1:IF Y=Y2 THEN LSTOP=1
       ELSE
                       X=X+1:IF X=X2 THEN LSTOP=1
               END IF
               JCOLN=POINT(X,Y)
       LOOP UNTIL JCOLN<>JCOLO         '===================================
       IF IDIR=%X THEN
               L=Y-YO
       ELSE
               L=X-XO
       END IF
       TOTALEN=TOTALEN+L
    IF JCOLO=PH1 OR JCOLO=BORDER THEN
               TOTLEN1=TOTLEN1+L
       ELSE
               TOTLEN2=TOTLEN2+L
       END IF
    IF LSTOP=1 THEN GOTO NextLine
       ISEG=ISEG+1
    IF JCOLO=PH1 THEN                      'come to border from PH1
        FLAG=1
    ELSEIF JCOLO=PH2 THEN                  'come to border from PH2
       FLAG=0
    END IF
    IF ISEG=1 THEN GOTO NextSegment
    IF JCOLO=PH2 THEN
               I2=I2+1
            L2(I2)=L
    END IF
    IF JCOLO=BORDER THEN                   'come from BORDER
        IF JCOLN=PH1 THEN
               IF FLAG THEN
               I2=I2+1
                L2(I2)=O
            END IF
        END IF
    END IF
       NextSegment:
       JCOLO=JCOLN
       XO=X:YO=Y
  LOOP   '**************************************************************
  NextLine:
  NLINES=NLINES+1
LOOP   '-----------------------------------------------------------------

AREA1=TOTLEN1/TOTALEN
AREA2=TOTLEN2/TOTALEN
areatot=area1+area2:print "areatot",areatot

END SUB

'==========================================================================
SUB OGIVE(XSCALE,YSCALE,NPOINTS,LDATAPT(),DNORMAL,OTITLE$())

'plots ogive of LDATAPT divided by DNORMAL
'with captions given by OTITLE$

LINE (363, 1)-(639, 479), 1, B
PAINT (460, 200), 3, 1
```

```
PAINT (460, 200), 0, 1
Checkgraph:
        LOCATE 1, 47: PRINT OTITLE$(1)        ' Graph Title
        LOCATE 2, 47: PRINT OTITLE$(2)        ' SubTitle1
        LOCATE 3, 47: PRINT OTITLE$(3)        ' Y-axis Caption
        LOCATE 4, 47: PRINT OTITLE$(4)        ' Sub Y-axis caption
        LOCATE 13, 47: PRINT "0.5"
        LOCATE 22, 47: PRINT "  0"
        LOCATE 23, 49: PRINT "  0"
        LOCATE 4, 47: PRINT "1.0"
        LOCATE 23, 58: PRINT "  1"
        LOCATE 23, 67: PRINT "   2"
        LOCATE 23, 76: PRINT "   3"
        LOCATE 24, 51: PRINT OTITLE$(5)       'X-axis Caption
        LOCATE 25, 51: PRINT OTITLE$(6)       'Sub X-axis caption
        LINE (400, 60)-(630, 340), 9, B
        PAINT (405, 65), 6, 9
        YORD = 280: XORD = 230
        XSCALE = 3: YSCALE = NPOINTS+1
        FOR ILINE = 0 TO XSCALE
                KXLINE1 = 400 + ILINE * XORD / XSCALE
                LINE (KXLINE1, 340)-(KXLINE1, 340-YORD), 9
        NEXT ILINE
        KYLINE1 = 400 / 2
        LINE (400, KYLINE1)-(630, KYLINE1), 9
        FOR KP = 1 TO NPOINTS
                KY = (KP / YSCALE) * YORD
        KX = (LDATAPT(KP) / (DNORMAL * XSCALE)) * XORD
                IF KP = 1 THEN
                        PSET (KX + 400, 340 - KY), 3
                ELSE
                        LINE STEP(0, 0)-(KX + 400, 340 - KY), 3
                END IF
        NEXT KP
END SUB
'========================================================================
SUB REALMICRO(AvgDiam,AreaFrac)

CLS
INPUT "Please enter .BIF filename (w/o extension): ", Infile$
Infile$=Infile$+".BIF"

'input file in BIF format; ASCII numbers represent fiber cross-
'section pixels of various colors (each fiber has unique color
'number (up to 256), except fibers intersecting image frame have
'color 10) against a background of color 0.

OPEN Infile$ FOR INPUT AS #1       'Opens the image file
CLS

BackGround=0                       'Initialize conditions and counters
EdgeFiberColor=10
NColorMx=256
NFullFibers=0
FullFibPixCount=0
AllFibPixCount=0
NPixY=479
NPixX=511

DIM Count(256)

FOR y = 0 TO NPixY                   '479 in VGA
FOR x = 0 TO NPixX                   '511 in VGA

        INPUT #1, Col                        'Gets data from file

    if Col<>BackGround and Col<>EdgeFiberColor then
        i=Col
        Count(i)=Count(i)+1                  'Counts pixels in
        FullFibPixCount=FullFibPixCount+1    'full-area fibers
        end if
```

```
        if col<>BackGround then
        Col=5
         PSET (x, y), Col                    'Puts data on screen
         AllFibPixCount=AllFibPixCount+1     'Counts all fiber pixls
        end if

NEXT x
NEXT y

for i=1 to NColorMx
        if Count(i)>0 then
        NFullFibers=NFullFibers+1          'Counts number of
        end if                                 'full-area fibers
next i

'------------------------------------Calculate statistics
AreaFib=FullFibPixCount/NFullFibers
AvgDiam=sqr(AreaFib*4/3.14159)
AreaTot=(NPixX+1)*(NPixY+1)
AreaFrac=AllFibPixCount/AreaTot
'------------------------------------Print if in checkout (KCO=1)
if kco=1 then
        PRINT "AVG FIBER AREA   " ;:PRINT USING "######.##";AreaFib
        PRINT "No. FULL FIBERS  " ;:PRINT USING "     ####";NFullFibers
        PRINT "EQUIV FIBER DIAM " ;:PRINT USING "######.##";AvgDiam
        PRINT "FIBER AREA FRACT " ;:PRINT USING "######.##";AreaFrac
'---------------------------------------------------------------
        circle (560,240),AvgDiam/2,7          'display equivalent circle
        paint (560,240),7,7
        locate 19,65:print " Equiv Circle"
end if

END SUB



'==========================================================================
'                                 NOTES

'Populates a rectangular area with reddish colored circles that do not
'overlap.  To produce a "natural" arrangement within the rectangle,
'circles are placed randomly outside as well as inside the rectangle
'but the ones outside are not counted as to any statistics.

'Randomly placed horizontal and vertical lines are scanned across the
'image.  The output is the length distribution of line segments that
'are entirely within the background phase (matrix) with endpoints at
'circle boundaries.

'Overlap of circles is allowed to the extent indicated by the input FDERR,
'the ratio of overlap to diameter.  If no input is provided, FDERR is taken
'as zero.  However, overlap of (up to) one pixel is built in as an
'adjustment to RFO, the radius of the exclusion circle; otherwise, contact
'would never occur.

'This version displays on a standard VGA screen.
'It is a PRELIMINARY version.
'Please report bugs to jjortner@oregoncoast.com or (503)392-4001.

'For illustrative purposes, if requested, the code will plot
'a regular hexagonal array, an open hexagonal array, or a regular
'square array, at an area fraction input by the user.  Otherwise the
'array is random.  The regular arrays can be rotated by a random
'angle, to make the scanning directions random with respect to
'the regular array.

'The attained area fraction of randomly placed circles depends, in
'a random sort of way, on the number of iterations requested before
'the program gives up on finding a space for the newest circle.  The
```

```
'maximum attainable random Af is about 0.6, at 5000 or more iterations.

'After circles are in, area fraction is estimated from a complete
'pixel-by-pixel scan of the counting space.  Also, area fraction is
'estimated from random point count, from a crude count of the number
'of full and partial circles in the counting box, and from the line-
'scan data (as the ratio of total length within circles to the total
'line length).

'For an area fraction equal to the pixel-by-pixel area-fraction estimate,
'the center-center distance and the proximity distance of a regular
'hexagonal array are calculated.  These hex-array parameters are used
'to normalize the segment lengths before they are plotted on a cumulative
'frequency ogive.

'The code shows some peculiar effects due to pixel reprentations of
'circles.  For example, the pixel area fraction for regular arrays is
'always larger than the input value (which is used as if the circles
'were geometrically perfect.  To reduce the effect on the normalization
'of statistics for regular arrays, the parameters for the equivalent
'hexagonal array are computed for a circle radius that is adjusted to
'reflect the pixel-induced error in circle area.  This adjustment to
'radius is printed out as part of the results.

'Another peculiarity is that, because of these pixel effects, the
'attained area fraction of regular arrays can be higher than
'theoretically possible.  Therefore, requesting regular hexagonal
'arrays of area fraction higher than about 0.84 leads to negative
'proximity distances and other nonsense.
```

'Program REALMIC Julius Jortner, Nov 95

```
'PROVIDES FOR READING REAL MICROGRAPHS................BIF FORMAT
'GIVES LINE-SEGMENT STATISTICS

'Allows reading in and analyzing up to about 8 bif images...
'which must have filenames XX_n.BIF that are in increasing order of n
'where n is an integer and is stepped by 1.

DEFINT I-N:
OPTION BASE 1
CLS
SCREEN 12


'===================================================
NM = 400: XMX = 505: YMX = 475: DB = 5

PRINT                                   '--------------------inputs
INPUT "SPECIAL PRINTOUTS, CHECKOUT ONLY? ENTER 1: ", kco
INPUT " Enter 1 if you want to pause between images: ", KPAUSE

DIM  lmatr(20000),lprox(20000),ldist(20000),OTITLE$(24),InFile$(30)

PI = 3.14159
'------------------------------------------------------------------
JCOLBOX = 10                            'color for counting_area box
JCOLCIRC = 5                            'color for fibers
JCOLCONT = 8                            'color for contact links

CLS

CALL FILEIN(Ncases,Istart,File$,Infile$())

'Outfile$ = File$ + ".OUT"
'OPEN Outfile$ FOR OUTPUT AS #2    'Opens a file for numerical output
cls
iseg=0
```

```
NSEG2=0

For k=1 to Ncases       '       ===================== Do designated cases

Inpf$=Infile$(k)+".bif"
OPEN Inpf$ FOR INPUT AS #1      'Opens the image file

LOCATE 1,68
PRINT Infile$(k)
'-----------------------------------------read .BIF image file & draw
Nstart=NSEG2+1
CALL REALMICRO(DF,AreaFrac,X1,X2,Y1,Y2,kco,iseg,lmatr(),NSEG2)
Nend=NSEG2
RF = DF / 2
if k=1 then RFBase=RF
RFSQ=RF^2+RFSQ
LINE (X1 - 1, Y1 - 1)-(X2 + 1, Y2 + 1), JCOLBOX, B
IF kco = 1 THEN
        CALL PIXAREA(X1, X2, Y1, Y2, JCOLCIRC, areapix)'area fraction via pixels
END IF
'----------------------------------------- calc stats for hexagonal array
CALL HEXSTATS(RF, AreaFrac, HEXPROX, DC)
'   print hexprox,dc,rf:stop
Nlen=Nend-Nstart+1
For il=Nstart to Nend
        lmatr(il)=lmatr(il)*RF/RFBase
'       lprox(il)=lmatr(il)/HEXPROX
'   ldist(il)=lmatr(il)/DC
'   print lmatr(il),lprox(il),ldist(il)
'   if il=20 then stop
Next il
'--------------------------------------------------'print image stats
LOCATE 19, 1
PRINT " ANALYSIS AND AREA FRACTIONS"
PRINT " Image file..........."; : PRINT USING "&"; InFile$(K)
PRINT " Line-seg Area Frac:  "; : PRINT USING "#.####"; AreaFrac
PRINT " # segmts in matrix : "; : PRINT USING "######"; NSEG2
IF kco = 1 THEN
        PRINT " pixel area fraction: "; : PRINT USING "#.####"; areapix
    AVGAPIX=AREAPIX+AVGAPIX
END IF
IF KPAUSE=1 OR k=NCases THEN
        WHILE INKEY$=""
                LOCATE 28,1
        PRINT " To continue, press any key..."
        WEND
END IF

AVGAREA=AREAFRAC+AVGAREA
NSEGTOT=NSEG2+NSEGTOT
CLOSE #1                              'closes InFile$(k)

Next k          '======================================

AreaFrac=AVGAREA/Ncases
RFSQ=RFSQ/Ncases
RF=sqr(RFSQ)
'----------------------------------------- calc stats for hexagonal array
CALL HEXSTATS(RFBase, AreaFrac, HEXPROX, DC)
'   print hexprox,dc,rf:stop


LOCATE 19, 1
PRINT " ANALYSIS AND AREA FRACTIONS"
PRINT " Image file series............ "; : PRINT USING "&"; File$
PRINT " Line-seg Area Frac:  "; : PRINT USING "#.####"; AreaFrac
PRINT " # segmts in matrix : "; : PRINT USING "######"; NSEGTOT
IF kco = 1 THEN
        PRINT " pixel area fraction: "; : PRINT USING "#.####"; AVGAPIX/NCASES
END IF
```

```
LOCATE 28,1
PRINT " RESULTS FOR IMAGES FROM FILES:     "
PRINT " ";:PRINT USING "&"; Infile$(1)," TO ";
PRINT USING "&"; Infile$(Ncases)

'CALL SORTARR(NSEGTOT, lmatr())          'sort LMATR in ascending order
array sort lmatr() for nsegtot
'array sort lprox() for nsegtot          'PowerBasic array sort
'array sort ldist() for nsegtot
'-------------------------------------------set up to print graph

        OTITLE$(1) = "Line-Segment Lengths for Matrix"
        OTITLE$(2) = "                               "
        OTITLE$(3) = "    CUMULATIVE FREQUENCY"
        OTITLE$(4) = "                    "
        OTITLE$(5) = " SEGMENT LENGTH RELATIVE"
OgiveAgain:

        OTITLE$(6) = "   TO MIN HEX PROXIMITY "
'    HEXPROX=1
        CALL OGIVE(XSCALE, YSCALE, NSEGTOT, lmatr(), HEXPROX, OTITLE$())

        WHILE INKEY$=""
                LOCATE 1,1
        PRINT " To see ogive normalized to reg-hex center distance"
       PRINT " press press any key..."
        WEND
        OTITLE$(6) = " TO HEX CENTER DISTANCE "
'    DC=1
        CALL OGIVE(XSCALE, YSCALE, NSEGTOT, lmatr(), DC, OTITLE$())
   INPUT " To END, press 'x'...", kend$
   if kend$="x" or inkey$="X" then goto Endloop
        WHILE INKEY$=""
                LOCATE 1,1
        PRINT " To see ogive normalized to reg-hex proximity distance"
       PRINT " press press any key except 'x'..."
   WEND
   INPUT " To END, press 'x' or continue with 'ENTER'...", kend$
   if inkey$="x" or inkey$="X" then goto Endloop
   PRINT " To continue, press ENTER..."
        goto OgiveAgain
Endloop:
IF INSTAT THEN END
GOTO Endloop

'-----------------------------------------------------END

'FUNCTIONS AND PROCEDURES
'==========================================================================
DEF FNHYPO (x, y) = SQR(x ^ 2 + y ^ 2)'calculates hypotenuse via Pythagoras
'=====================================================================

'NOTES

'Reads an image file in BIF format (Ben Yurgartis, 10/95) and displays
'it in one color for fibers and another for background.

'All horizontal pixel lines are scanned across the image.
'The output is the length distribution of line segments that
'are entirely within the background phase (matrix) with endpoints at
'fiber boundaries.  Also included in the matrix length statistics are
'zero lengths where the line crosses contacts between two fibers.
'(First and last line segments in each scan are ignored to avoid
' including data from partial fibers and incomplete matrix regions
' at the edges of the image area).

'The area fraction of fibers is also calculated from the line scans.

'The length statistics are normalized by certain dimensions of a
'regular hexagonal array of circular fibers of the same area fraction.
'Specifically, the user can choose to normalize to the hex-array
```

```
'center-center distance or to the hex-array min. proximity distance.


'This version displays on a standard VGA screen.
'It is a PRELIMINARY version.
'Please report bugs to jjortner@oregoncoast.com or (503)392-4001.
'===============================================================



'==================================================================
SUB HEXSTATS (R, AF, HEXPROX, DC)

'calculates the closest perimeter-perimeter distance HEXPROX
'and the center-center distance DC between circles of radius R
'arranged in a regular hexagonal array of area fraction AF

        PI = 3.14159
        ACIRC = PI * R ^ 2
           DF = 2 * R
        AHEXAGON = 3 * ACIRC / AF
        ANGLE = 30 * PI / 180: COSFACT = COS(ANGLE)
        DC = SQR(AHEXAGON / (3 * COSFACT))
           IF DC < DF THEN
                PRINT "Requested Area Fraction Too High!: ", AF: END
           END IF
           HEXPROX = (DC - DF)

END SUB

'==================================================================
SUB OGIVE (XSCALE, YSCALE, NPOINTS, LDATAPT(), DNORMAL, OTITLE$())

'plots ogive of LDATAPT divided by DNORMAL
'with captions given by OTITLE$

LINE (363, 1)-(639, 479), 1, B
PAINT (460, 200), 3, 1
PAINT (460, 200), 0, 1
Checkgraph:
        LOCATE 1, 47: PRINT OTITLE$(1)       ' Graph Title
        LOCATE 2, 47: PRINT OTITLE$(2)       ' SubTitle1
        LOCATE 3, 47: PRINT OTITLE$(3)       ' Y-axis Caption
        LOCATE 4, 47: PRINT OTITLE$(4)       ' Sub Y-axis caption
        LOCATE 13, 47: PRINT "0.5"
        LOCATE 22, 47: PRINT "  0"
        LOCATE 23, 49: PRINT "  0"
        LOCATE 4, 47: PRINT "1.0"
        LOCATE 23, 58: PRINT "  1"
        LOCATE 23, 67: PRINT "   2"
        LOCATE 23, 76: PRINT "   3"
        LOCATE 24, 51: PRINT OTITLE$(5)      'X-axis Caption
        LOCATE 25, 51: PRINT OTITLE$(6)      'Sub X-axis caption
        LINE (400, 60)-(630, 340), 9, B
        PAINT (405, 65), 0, 9
        YORD = 280: XORD = 230
        XSCALE = 3: YSCALE = NPOINTS + 1
        FOR ILINE = 0 TO XSCALE
                KXLINE1 = 400 + ILINE * XORD / XSCALE
                LINE (KXLINE1, 340)-(KXLINE1, 60), 9
        NEXT ILINE
        KYLINE1 = 400 / 2
        LINE (400, KYLINE1)-(630, KYLINE1), 9
        FOR KP = 1 TO NPOINTS
                KY = (KP / YSCALE) * YORD
        KX = (LDATAPT(KP) / (DNORMAL * XSCALE)) * XORD
                IF KP = 1 THEN
                        PSET (KX + 400, 340 - KY), 3
                ELSE
                        LINE STEP(0, 0)-(KX + 400, 340 - KY), 3
                END IF
```

```
        NEXT KP
END SUB

'=================================================================
SUB PIXAREA (X1, X2, Y1, Y2, JCOLCIRC, AF)
'estimate area fraction AF of a phase colored JCOL
'within a rectangular area defined by (X1,Y1)-(X2,Y2)
'using pixel counting over a full scan of the area

KX1 = X1: KX2 = X2: KY1 = Y1: KY2 = Y2
AREAF = 0
AREAT = (X2 - X1 + 1) * (Y2 - Y1 + 1)
FOR KX = KX1 TO KX2
           FOR KY = KY1 TO KY2
                IF POINT(KX, KY) = JCOLCIRC THEN
                           AREAF = AREAF + 1
                END IF
           NEXT KY
NEXT KX
AF = AREAF / AREAT
END SUB

'============================================================
SUB REALMICRO (AvgDiam,AreaFrac,X1,X2,Y1,Y2,kco,k,lmatr(),NSEGM)

'adapted from routine written by Ben Yurgartis, to
'acquire image from file in BIF format

'BIF format:               ASCII numbers represent fiber cross-
'section pixels of various colors (each fiber has unique color
'number (up to 256), except fibers intersecting image frame have
'color 10) against a background of color 0.

'CLS

BackGround = 0                   'Initialize conditions and counters
EdgeFiberColor = 10
NColorMx = 256
NFullFibers = 0
FullFibPixCount = 0
AllFibPixCount = 0
NPixY = 479
NPixX = 511

DIM Count(256)
'-----------------------------------------------------------------
NB = 4      'set up to strip blank border from image file

NCountX1 = NB: NCountY1 = NB: NCountX2 = NPixX - NB: NCountY2 = NPixY - NB
NCountX = NCountX2 - NCountX1 + 1: NCountY = NCountY2 - NCountY1 + 1
X1 = NCountX1: X2 = NCountX2: Y1 = NCountY1: Y2 = NCountY2
LINE (NCountX1, NCountY1)-(NCountX2, NCountY2), 6, B
'-----------------------------------------------------------------
KIN=K
FOR y = 0 TO NPixY                'scan image file - 479 in VGA
ISeg = 0
FOR x = 0 TO NPixX                'scan image file - 511 in VGA

  INPUT #1, Col                     'Gets data from file

  IF x >= NCountX1 AND x <= NCountX2 AND y >= NCountY1 AND y <= NCountY2 THEN
    IF x = NCountX1 THEN
        jOldCol = Col
           xold = x
    ELSEIF x < NCountX2 THEN
           jNewCol = Col
           IF jNewCol <> jOldCol THEN
                  ISeg = ISeg + 1
                xnew = x
                  IF ISeg = 1 THEN GOTO NextSegment
                  IF jOldCol = BackGround THEN
```

```
                    k = k + 1
                 lmatr(k) = xnew - xold
                ELSEIF jNewCol <> BackGround THEN
                     k = k + 1
                      lmatr(k) = 0
                END IF
NextSegment:
                 jOldCol = jNewCol
                xold = x
           END IF
    END IF

    IF Col <> BackGround AND Col <> EdgeFiberColor THEN
       i = Col
          Count(i) = Count(i) + 1                'Counts pixels in
          FullFibPixCount = FullFibPixCount + 1 'full-area fibers
          NBack = 0
       END IF

       IF Col <> BackGround THEN
       Col = 5
              AllFibPixCount = AllFibPixCount + 1'Counts all fiber pixls
       ELSE
       Col = 7
    END IF
    PSET (x, y), Col                   'Puts data on screen
  END IF
NEXT x
NEXT y
NSEGM = k-kin

FOR i = 1 TO NColorMx
       IF Count(i) > 0 THEN
       NFullFibers = NFullFibers + 1      'Counts number of
       END IF                                 'full-area fibers
NEXT i

'-------------------------------------Calculate statistics
AreaFib = FullFibPixCount / NFullFibers
AvgDiam = SQR(AreaFib * 4 / 3.14159)
AreaTot = CSNG(NCountX) * CSNG(NCountY)
AreaFrac = AllFibPixCount / AreaTot
'-------------------------------------Print if in checkout (KCO=1)
IF kco = -1 THEN
       PRINT "AVG FIBER AREA    "; : PRINT USING "######.##"; AreaFib
       PRINT "No. FULL FIBERS   "; : PRINT USING "     ####"; NFullFibers
       PRINT "EQUIV FIBER DIAM "; : PRINT USING "######.##"; AvgDiam
       PRINT "FIBER AREA FRACT "; : PRINT USING "######.##"; AreaFrac
'---------------------------------------------------------------
       CIRCLE (560, 240), AvgDiam / 2, 7     'display equivalent circle
       PAINT (560, 240), 7, 7
       LOCATE 19, 65: PRINT " Equiv Circle"
    stop
END IF

END SUB

'=====================================================================
SUB SORTARR (N, LRA())
'a heap sort algorithm from Numerical Recipes, Press et al
'1986 edition.  LRA is the input array of which the first
'N elements are sorted in ascending order and sorted LRA is
'output.

L = N / 2 + 1
IR = N

DO
       IF L > 1 THEN
       L = L - 1
        RRA = LRA(L)
```

```
      ELSE
      RRA = LRA(IR)
       LRA(IR) = LRA(1)
       IR = IR - 1
       IF IR = 1 THEN
             LRA(1) = RRA
          EXIT SUB
             END IF
  END IF
  i = L
  J = L + L
Again:
  IF J <= IR THEN
      IF J < IR THEN
             IF LRA(J) < LRA(J + 1) THEN J = J + 1
       END IF
       IF RRA < LRA(J) THEN
             LRA(i) = LRA(J)
          i = J
          J = J + J
       ELSE
             J = IR + 1
       END IF
      GOTO Again
  END IF
  LRA(i) = RRA
LOOP
END SUB

'=====================================================================
SUB FILEIN(Ncases,Istart,Inf$,Infile$())

'reads in a series of up to 30 filenames of the type "XX_n.BIF"
'where XX is an input string, n is a number, and BIF
'indicates the filename extension.
'The first file will have n=Istart. Others will have
'an n incremented by 1, until n=Istart+Ncases


CLS
Locate 2,1
PRINT "Please designate the file series."
PRINT "All files are named 'XY_n.BIF'."
PRINT "where n is a number between 1 and 30.
PRINT
PRINT "NOTE: XY should include the DOS Path"
PRINT
INPUT "Please enter the series name (XY): ", Inf$
INPUT "What is the first value of n ? : ", Istart
INPUT "How many files do you want ? :", Ncases
CLS
i=Istart
For j=1 to Ncases
If i<10 then
        Infile$(j)=Inf$ + "_" + STRING$(1,i+48)
ElseIf i>9 and i< 20 then
        Infile$(j)=Inf$ + "_" + STRING$(1,49)+STRING$(1,(i-10)+48)
ElseIf i>19 and i<30 then
        Infile$(j)=Inf$ + "_" + STRING$(1,50)+STRING$(1,(i-20)+48)
ElseIf i>29 and i<40 then
        Infile$(j)=Inf$ + "_" + STRING$(1,51)+STRING$(1,(i-30)+48)
End If

'print Infile$(j)

i=i+1
Next j

END SUB
```